Multiple Instance Learning Under Real-World Conditions

by

Marc-André CARBONNEAU

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN PARTIAL FULFILLMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph. D.

MONTREAL, OCTOBER 3, 2017

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS:

Mr. Ghyslain Gagnon, Thesis Supervisor
Department of Electrical Engineering, École de technologie supérieure

Mr. Eric Granger, Thesis Co-supervisor
Department of Automated Manufacturing Engineering, École de technologie supérieure

Mr. Stéphane Coulombe, President of the Board of Examiners
Department of Software and IT Engineering, École de technologie supérieure

Mr. Marco Pedersoli, Member of the jury
Department of Automated Manufacturing Engineering, École de technologie supérieure

Mr. Marco Loog, External Examiner
Pattern Recognition Laboratory, Delft University of Technology

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON SEPTEMBER 11, 2017

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

First, I would like to express my gratitude to my supervisors. Thank you for the moral and financial support. Thank you for your guidance. Thank you for the multiple teaching and work opportunities. Thank you for patiently reading each and every paper as I painfully learned to write in English. Thanks for the debates. Thank you for your time. Thank you for your friendship.

I am grateful to my family. Each one of you has been supportive from day one. Your encouragements helped me believe that this extended stay at school made some sense. I'm lucky and proud to be part of this exclusive club and love each one of you more than anything in the world.

Next, I need to thank all of these close friends that were and are there for me. Special mention to the BGNers, les voisines and Pélippe.

Finally, thank you to all my colleagues at LIVIA, LACIME and the unaffiliateds for making ÉTS a fun place to study.

APPRENTISSAGE PAR INSTANCES MULTIPLES DANS DES CONDITIONS RÉELLES

Marc-André CARBONNEAU

RÉSUMÉ

L'apprentissage par instances multiples (AIM) est un type d'apprentissage machine avec faible supervision. Les données sont groupées en ensembles que l'on nomme sacs. Une étiquette est donnée pour chacun des sacs. Par contre, les données individuelles dans les sacs, appelées instances, ne sont pas étiquetées. Comme pour les autres types d'apprentissages faiblement supervisés, l'AIM est utile quand il est coûteux même impossible d'obtenir des étiquettes pour chacune des instances. Dans tous les cas, on apprendra à partir de données arrangées en sacs. Cependant, la tâche du classificateur peut être de prédire la classe des sacs ou des instances. Cette formulation se révèle utile dans plusieurs situations passant de la prédiction des effets de médicaments à la reconnaissance visuelle d'objets. De par leur forme particulière, les problèmes d'AIM comportent plusieurs difficultés qui sont trop souvent mal comprises ou inconnues. Il en résulte que plusieurs méthodes AIM sont mal adaptées aux données réelles et présentent des performances inégales dependant des applications.

Dans cette thèse, des algorithmes de classification par AIM seront proposés pour la classification de sacs et d'instances, et ce, selon différentes suppositions sur les données. Chacune de ces méthodes est conçue pour être utilisée dans des situations réelles comportant des caractéristiques et défis particuliers. Comme première contribution, ces caractéristiques propres à l'AIM seront analysées et groupées en quatre catégories: le niveau auquel les prédictions sont faites, la composition des sacs, les types de distribution de données et l'ambiguïté sur les étiquettes. Chacune de ces catégories sera analysée en profondeur et les méthodes de pointe proposées pour ces cas spécifiques seront recensées. Ensuite, les applications typiques de l'AIM seront revues du point de vue de ces caractéristiques. Des expériences sont menées afin de montrer comment les caractéristiques affectent les performances de 16 types de méthodes d'AIM. Ces expérimentations et analyses permettent de tirer plusieurs conclusions pour choisir et tester des méthodes par AIM. Finalement, plusieurs sujets pour des recherches futures sont identifiés.

La seconde contribution est une méthode pour la classification de sacs basée sur l'identification probabiliste d'instances positives dans la base d'entraînement. Suite à ce processus d'identification, on entraîne un ensemble de classificateurs pour la classification d'instances. Les prédictions faites sur les instances sont ensuite combinées pour prédire la classe des sacs. Pour l'identification des instances positives, les données sont projetées dans plusieurs sous-espaces aléatoires. Dans ces sous-espaces, les instances sont regroupées et les étiquettes de sacs dans chaque groupe sont utilisées pour juger de la nature des instances. Les expériences montrent que cet algorithme obtient des performances comparables à l'état de l'art tout en étant davantage robuste à plusieurs des caractéristiques identifiées au chapitre précédent.

Il existe des applications pour lesquelles les instances ne peuvent pas être attribuées à une classe positive ou négative. En fait, les classes des sacs dépendent de la composition de ceux-

ci. Dans ces cas-là, ce sont les relations entre les instances qui portent l'information permettant de distinguer entre les classes de sacs. À titre de troisième contribution, une méthode pour la classification de sacs dans ces conditions est proposée. La méthode sert à prédire la personnalité d'un locuteur à partir de la voix. Cette méthode représente le spectrogramme d'un segment audio par un sac d'instances. Les parties du spectrogramme correspondent aux instances et sont encodées en utilisant un encodage creux (*sparse*). Une fois encodées, les instances sont agglomérées pour obtenir un vecteur de caractéristiques unique représentant le segment audio en entier. Ces vecteurs de caractéristiques sont utilisés par le classificateur de sac. Des expériences utilisant des données réelles montrent que la méthode obtient des résultats comparables à l'état de l'art tout en étant moins complexe à implémenter que les méthodes couramment utilisées dans le domaine.

Finalement, deux méthodes sont proposées pour choisir des sacs à faire étiqueter par un oracle dans un contexte d'apprentissage actif. Le but de l'apprentissage actif est d'entraîner un classificateur fiable en utilisant un minimum de données étiquetées. La structure des données en sacs rend sous-optimales les méthodes proposées pour l'apprentissage à instance simple. Les deux méthodes proposées tiennent compte de la structure en sacs mais abordent le problème différemment. La première tente de raffiner directement la frontière de décision du classificateur en portant son attention sur les instances près de celle-ci. La seconde méthode étudie la structure des instances dans l'espace afin d'identifier les régions les plus informatives. Le degré de désaccord entre les étiquettes des instances et des sacs et la proportion d'instances dont la classe est inconnue dans une région servent à déterminer la pertinence de celle-ci. Des expériences sont conduites dans un contexte d'apprentissage par induction et transduction pour trois domaines d'application. Ces expériences montrent la nécessité de considérer la structure en sacs dans un contexte d'AIM en réduisant la quantité d'étiquettes nécessaires pour l'obtention de bonnes performances de classification.

Cette thèse démontre que les problèmes d'AIM comportent une grande variété de défis et problématiques. Après une analyse en profondeur de ces défis et problématiques, des expériences sont menées afin de mesurer leur impact sur les performances des méthodes AIM. Ensuite, des méthodes sont proposées spécialement pour solutioner certaines de ces problématiques. Les méthodes sont validées expérimentalement avec des données provenant d'applications réelles. Finalement, des avenues pour recherches futures sont identifiées.

Mots clés: Apprentissage par instances multiples, apprentissage faiblement supervisé

MULTIPLE INSTANCE LEARNING UNDER REAL-WORLD CONDITIONS

Marc-André CARBONNEAU

ABSTRACT

Multiple instance learning (MIL) is a form of weakly-supervised learning that deals with data arranged in sets called bags. In MIL problems, a label is provided for bags, but not for each individual instance in the bag. Like other weakly-supervised frameworks, MIL is useful in situations where obtaining labels is costly. It is also useful in applications where instance labels cannot be observed individually. MIL algorithms learn from bags, however, prediction can be performed at instance- and bag-level. MIL has been used in several applications from drug activity prediction to object localization in image. Real-world data poses many challenges to MIL methods. These challenges arise from different problem characteristics that are sometimes not well understood or even completely ignored. This causes MIL methods to perform unevenly and often fail in real-world applications.

In this thesis, we propose methods for both classification levels under different working assumptions. These methods are designed to address challenging problem characteristics that arise in real-world applications. As a first contribution, we survey these characteristics that make MIL uniquely challenging. Four categories of characteristics are identified: the prediction level, the composition of bags, the data distribution types and the label ambiguity. Each category is analyzed and related state-of-the-art MIL methods are surveyed. MIL applications are examined in light of these characteristics and extensive experiments are conducted to show how these characteristics affect the performance of MIL methods. From these analyses and experiments, several conclusions are drawn and future research avenues are identified.

Then, as a second contribution, we propose a method for bag classification which relies on the identification of positive instances to train an ensemble of instance classifiers. The bag classifier uses the predictions made on instances to infer bag labels. The method identifies positive instances by projecting the instances into random subspaces. Clustering is performed on the data in these subspaces and positive instances are probabilistically identified based on the bag label of instances in clusters. Experiments show that the method achieves state-of-the-art performance while being robust to several characteristics identified in the survey.

In some applications, the instances cannot be assigned to a positive or negative class. Bag classes are defined by a composition of different types of instances. In such cases, interrelations between instances convey the information used to discriminate between positive and negative bags. As a third contribution, we propose a bag classification method that learns under these conditions. The method is a applied to predict speaker personality from speech signals represented as bags of instances. A sparse dictionary learning algorithm is used to learn a dictionary and encode instances. Encoded instances are embedded in a single feature vector summarizing the speech signal. Experimental results on real-world data reveal that the proposed method yields state-of-the-art accuracy results while requiring less complexity than commonly used methods in the field.

Finally, we propose two methods for querying bags in a multiple instance active learning (MIAL) framework. In this framework the objective is to train a reliable instance classifier using a minimal amount of labeled data. Single instance methods are suboptimal is this framework because they do not account the bag structure of MIL. The proposed methods address the problem from different angles. One aims at directly refining the decision boundary, while the other leverage instance and bag labels to query instances in the most promising clusters. Experiments are conducted in an inductive and transductive setting. Results on data from 3 application domains show that leveraging bag structure in this MIAL framework is important to effectively reduce the number of queries necessary to attain a high level of classification accuracy.

This thesis shows that real-world MIL problems pose a wide range of challenges. After an in-depth analysis, we show experimentally that these challenges have a profound impact on the performance of MIL algorithms. We propose methods to address some of these challenges and validate them on real-world data sets. We also identify future directions for research and remaining open problems.

Keywords: Multiple-Instance Learning, Random Subspace Methods, Weakly Supervised Learning, Classification, Active Learning, Personality Prediction

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABREVIATIONS

AGIN	AGgregated INformativeness
AL	Active Learning
APR	Axis-Parallel Rectangle
AUC	Area Under the ROC Curve
AUC_{PR}	Area Under the Precision-Recall Curve
BoW	Bag-of-Words
C-BAS	Cluster-Based Aggregative Sampling
CAD	Computer-Assisted Diagnosis
CBIR	Content-Based Image Retrieval
CCE	Constructive Clustering-based Ensemble
CkNN	Citation k-Nearest Neighbors
DD	Diverse Density
EM-DD	Expectation Maximization Diverse Density
EMD	Earth-Mover's Distance
KI-SVM	Key Instance SVM
MI	Multiple Instance
MI-SVM	MI SVM algorithm
mi-SVM	mixed-integer SVM algorithm
MIAL	Multiple Instance Active Learning

miGraph	MI Graph Algorithm
MIL	Multiple Instance Learning
MILES	MIL via Embedded instance Selection
MInD	Multiple Instance Dissimilarity
NAULC	Normalized Area Under the Learning Curve
NSK	Normalized Set Kernel
QP	Quadratic Programming
RBF	Radial Basis Function
ROC	Receiver Operating Characteristic
RSIS	Random Subspace Instance Selection
RSWI	Random Subspace Witness Identification
sbMIL	sparse balanced MIL
SDB-MIL	Sphere-Description-Based MIL
SIL	Single Instance Learning
SIVAL	Spatially Independent, Variable Area, and Lighting
sMIL	sparse MIL
SMILe	Shuffled MIL
stMIL	sparse transductive MIL
SVM	Support Vector Machine
UAR	Unweighted Average Recall
WR	Witness Rate

INTRODUCTION

In recent years, technological developments have allowed to generate large quantities of data for various applications ranging from computer-aided diagnosis in health care to sentiment analysis in natural language. While data is available, learning predictive models from it raises different challenges. Complete annotations must be provided for each data entry in fully-supervised learning. However, annotating data is costly in terms of time and resources. In most cases, it involves one or more human annotators examining data points one by one to provide labels and, sometimes, the location of regions of interest. For example, modern large scale data sets such as ImageNet (Russakovsky *et al.*, 2015) or MS-COCO (Lin *et al.*, 2014) contain hundreds of thousands of images with local annotation with bounding boxes or segmentation for objects. Collecting annotations for this amount data is a colossal enterprise which required an equivalently colossal workforce (Russakovsky *et al.*, 2015). In medical imaging and affective computing applications, annotations are made by a committee of domain experts which also incurs high costs. This is why learning frameworks that alleviate the burden of annotation, such as semi-supervised, active and weakly supervised learning, are receiving much attention from the machine learning community. This is one of this thesis motivations for studying Multiple Instance Learning (MIL), which is a form of weakly supervised learning.

With MIL, objects are represented by a collection of parts. A collection is usually called a bag and each individual part is called an instance. A label is provided for a bag, but not for individual instances. Figure 0.1 illustrates an example of a visual object detection problem formulated as a MIL classification problem. Here, the objective is to train a classifier to detect coffee mugs. Each image is a bag. The segments of the image correspond to instances. Weak supervision is provided by bag labels: a bag belongs to the positive class if the image contains a coffee mug, otherwise, it belongs to the negative class. Traditional fully-supervised learners would require a bounding box indicating the position of the coffee mug in the images to learn properly. If the learner were to be trained from the whole image, the other objects in the background would

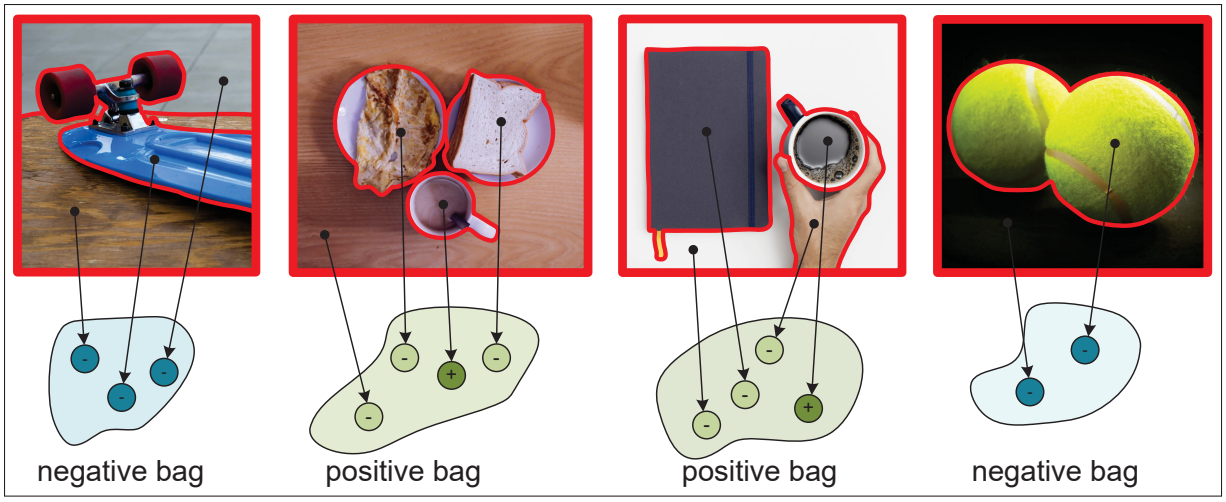


Figure 0.1 Example of a MIL problem where the objective is to recognize images containing a coffee mug

also be considered as coffee mugs which would degrade recognition performance. In contrast, in this situation, a MIL learner would disambiguate the nature of each instance to train the detector.

Motivations for MIL

Learning a recognition model from whole image labels, without local annotation, is useful in several application domains. For instance, in (Xu *et al.*, 2016; Karpathy & Fei-Fei, 2015; Fang *et al.*, 2015) the system learns to detect objects in images from words in captions. In (Zhu *et al.*, 2015), the system learns from images returned by queries on web search engines. No annotators are needed since the bag labels are simply query words entered in the search engine. MIL is also increasingly employed in medical imaging applications (Quellec *et al.*, 2016). In this context, the MIL framework is attractive because the system can learn from the diagnosis of a patient without local annotation from experts. This means that a larger quantity of data can be leveraged for training computer-assisted diagnosis systems (CAD). Moreover, it has been shown that in some cases, MIL systems outperformed fully supervised systems (Quellec *et al.*, 2017). Some image acquisition technologies make it difficult for experts to accurately identify

and segment all target patterns because of the lack of clear object contours. In these situations, it is better to let the learning system manage this problem (Quellec *et al.*, 2016). Also, there are some general cues in images which may not be isolated – or might be unknown to clinicians. This means that traditional classifiers for single instance learning (SIL) cannot take advantage of these cues.

Learning from weakly supervised data with MIL is not limited to visual data. The methods proposed for text data in (Kotzias *et al.*, 2015), predicts the sentiment of individual sentences using the overall rating associated with user-reviews. In this example, sentences are instances and complete reviews are bags. In (Briggs *et al.*, 2012) a bird song classifier is trained using audio recordings from unattended microphones in the wilderness. A recording contains various species of birds. For a given species, a recording corresponds to a positive bag if the microphone is placed in a region where the species can be encountered.

Another motivation for MIL, aside from the ability to learn from weak labels, is that some problems cannot be formulated as traditional SIL problems. In fact, this is what initially lead to the proposal of the MIL framework in (Dietterich *et al.*, 1997). This seminal paper studies the problem of drug activity prediction. The objective is to predict if a molecule will induce a target effect. A molecule can take many conformations (i.e. atom arrangements). These conformations cannot be produced in isolation. This means that when testing a given molecule, the effects of many conformations are observed at the same time. If the target effect is induced, some conformations might be inactive, but at least one of them is active. On the other hand, if a molecule does not induce the target effect, all of its conformations are inactive. If a molecule is modeled as a bag and the conformations as instances, this problem corresponds to the *standard MIL assumption* (Foulds & Frank, 2010). In this problem, instances cannot be observed individually for technological reason. In other problems, it is not possible to label instance because of limited knowledge. In that case, MIL can be used to discover which instances cause an ob-

served effect to help researchers better understand a phenomenon. For example, in (Palachanis, 2014), the genomic features governing the bonding of transcription factors in gene expression are discovered using MIL. Bags represent genes, and transcription factors are instances. By comparing expressed genes with their counterparts, the responsible transcription factors were discovered.

Finally, in some applications, an object is a composition of different parts which do not define class membership when considered individually. For these problems, the standard MIL assumption is relaxed to the collective MIL assumption (Foulds & Frank, 2010). For example, in the Bag-of-Word (BoW) model (Harris, 1954), texts are described as collections of words. Each word is not enough to predict the subject of a text. However, when all words are considered together their relations carry significant information. This model can be applied to visual data for content-based image retrieval (CIBR) (Csurka *et al.*, 2004) by replacing words by visual key-points.

MIL Classification

In MIL, there are two types of classification problems: instance-level and bag-level classification. These two tasks, while related, are different. In both cases, the classifier is trained with MIL data. However the granularity of the prediction is different. In instance-level classification tasks, the objective is to predict each instance label. In contrast, in bag-level classification tasks discovering the exact label of each instance is not that important, as long as the correct bag label is predicted.

Traditionally MIL research has focused on bag-level classification. This type of problem can be approached from 2 different angles (Amores, 2013). One possible approach is to reason in bag-space. Bags can be compared directly using set distance metrics. Alternatively, the content of bags can be summarized in a single feature vector which transforms the MIL problem into

a supervised problem. The other way of approaching bag classification is to classify each instance individually and then, combine predictions to infer the label of the bag.

More recently, instance-level classification attracted attention. As will be shown later in the thesis, an instance classifier trained for bag-level classification is different from an instance classifier used for instance-level classification because misclassification costs are different.

Challenges of MIL in Real-World Applications

Using MIL in real-world applications is challenging. First, the degree of supervision entails uncertainty on instance classes. Depending on the working assumption, this uncertainty can be asymmetric. For example, under the standard MIL assumption, only instances in positive bag labels are ambiguous. In other cases, the label space for instance is different from the label space for bags. In instance classification problems, the ambiguity on the true instance labels makes it difficult to constitute a noise-free training set. Also, for the same reason, it is difficult to directly use instance classes in the cost function when training classifiers.

Secondly, MIL deals with problems where data is structured in sets (i.e. bags). Aside from set membership, this structure can have implications on how instances relate to each other. For example, some instances may co-occur more often in bags of a given class. Discriminative information may lie in these co-occurrences. In that case, the distribution of instances in bags must be modeled. Sometimes, instances of the same bag share similarities which are not shared with instances from other bags. A successful MIL method must be able to discover what information is related to class membership and not bag membership. Sometimes, there are very few positive instances in positive bags, which makes it difficult for the learner to identify them. These relations and their implications will be discussed in detail in Chapter 1.

Finally, MIL is often associated with class imbalance, especially with instance-level classification. Negative bags only contain negative instances while positive bags contain negative and

positive instances. Even with an equal number of bags in each class, there are more negative instances in the training set. This problem is more severe when only a small proportion of instances are positive in positive bags.

A lot of MIL methods make implicit assumptions about the data that are often violated in practice. This leads to disappointing results in real-world applications. For example, methods like Expectation Maximization Diverse Density (EM-DD) (Zhang & Goldman, 2001) and Sphere-Description-Based MIL (SDB-MIL) (Xiao *et al.*, 2016) assume that positive instances form a single cluster in feature space. Other methods such as Normalized Set Kernels (NSK) (Gärtner *et al.*, 2002) assume that positive bags contain a majority of positive instances. Methods using distance measures like Citation-kNN (CkNN) (Wang & Zucker, 2000) or Constructive Clustering-based Ensemble (CCE) (Zhou & Zhang, 2007) assume that every instance feature is relevant and that the location of an instance in the input space is mainly dependent on its class and not its bag membership.

Research Objectives and Contributions

In this thesis, we study MIL in challenging environments of real-world problems. MIL problems are often very different from one another because the aforementioned challenges arise at various degree. As a results, MIL methods may yield a high level of performance for an application, while being inappropriate for another. We first study what are the characteristics of MIL that influence performance of algorithms and how they relate to different application fields. Then, we propose methods able to cope with the challenges associated with these problem characteristics of real-world problems. Two methods are proposed for bag classification under different working assumptions and the instance classification task is addressed in an active learning framework.

There are six main contributions in this work which led to three journal and three conference publications:

a. A survey paper in which important problem characteristics for MIL are identified and categorized. Applications are analyzed in light of these characteristics and extensive experiments are conducted to measure their impact (see Chapter 1).

   Related publication:

   Multiple Instance Learning: A Survey of Problem Characteristics and Applications. *(In second round of revision in Elsevier's Pattern Recognition, 2017)*

b. A new method is proposed to identify positive instances in MIL data sets. The method relies on projecting the data into different random subspaces and cluster characterization. It is robust to many of the challenges posed by the problem characteristics identified in the survey (see Chapter 2 and Annex I).

   Related publications:

   Robust Multiple-Instance Learning Ensembles Using Random Subspace Instance Selection *(published in Elsevier's Pattern Recognition, 2016)*

   Witness Identification in Multiple Instance Learning Using Random Subspaces. *(published in the proceeding of the 23rd International Conference on Pattern Recognition (ICPR), 2016)*

c. A new bag classification method is proposed based on probabilistic positive instance identification. The probabilistic instance labels are used to sample training sets which, in turn, are used to build an ensemble of classifiers (see Chapter 2).

   Related publication:

   Robust Multiple-Instance Learning Ensembles Using Random Subspace Instance Selection *(published in Elsevier's Pattern Recognition, 2016)*

d. A bag-level method is proposed for the prediction of personality from the spectrogram of speech signals. The proposed framework is inspired from the BoW model in which features are learned from the data (see Chapter 3).

   Related publication:

   Feature Learning from Spectrograms for Assessment of Personality Traits. *(In second round of revision in IEEE Transactions on Affective Computing, 2016)*

e. Two query strategies are proposed to train a MIL instance classifier in an active learning framework. These methods leverage the bag structure of the data to guide an efficient exploration of the instance space (see Chapter 4).

   Related publication:

   Bag-Level Aggregation for Multiple Instance Active Learning in Instance Classification Problems. *(submitted to IEEE Transactions on Neural Networks and Learning Systems, 2017)*

f. A strategy to adapt bag-level classifiers to the instance-level classification task. This is achieved by adjusting the decision threshold on the score function learned by bag classifiers (see Annex II).

   Related publication:

   Decision Threshold Adjustment Strategies for Increased Accuracy in Multiple Instance Learning *(published in the proceeding the 6th International Conference on Image Processing Theory, Tools and Applications (IPTA), 2016)*

Additional contributions were made in computer vision and signal processing that led to the publication of a journal paper and a conference paper:

a. Detection of Alarms and Warning Signals on an Digital In-Ear Device. *(published in International Journal of Industrial Ergonomics, 2013)*

b. Real-Time Visual Play-Break Detection in Sport Events Using a Context Descriptor. (published in the IEEE International Symposium on Circuits and Systems (ISCAS), 2015)

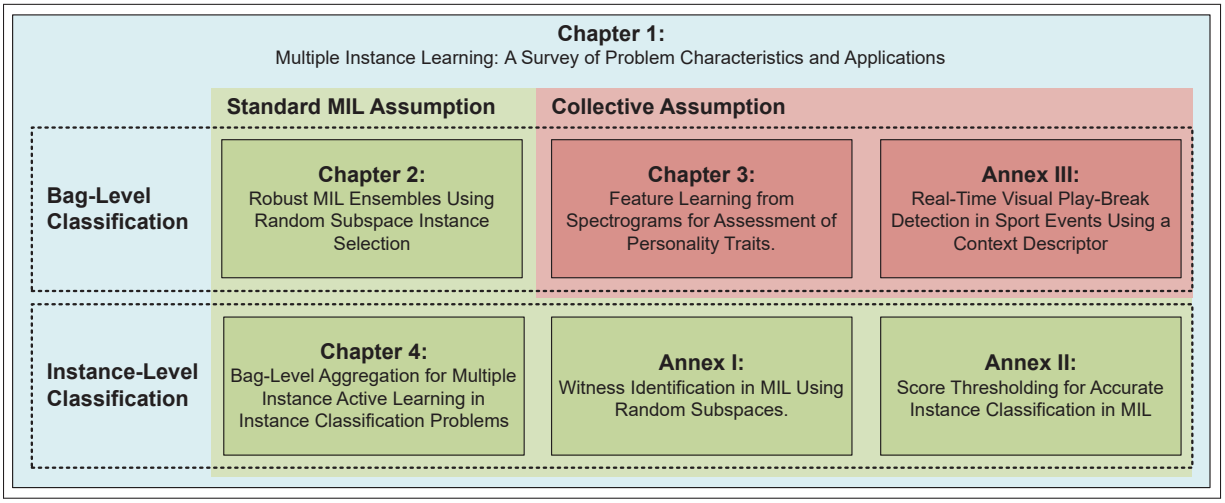


Figure 0.2 Overview of the thesis organization

Thesis Organization

This is a thesis by article, therefore each chapter in the main body corresponds to a publication. As a complement, the annexes contain other published articles that make additional related contributions. Figure 0.2 shows the relationship between each chapter and annex according to MIL assumptions and tasks. In Chapter 1, the literature review, the tasks, assumptions and challenges associated with MIL are surveyed and rigorously analyzed. It is explained that instance-level and bag-level classification are different tasks and that specific methods need to be used for each. Bag-level classification can be performed under different assumptions depending on the application. In the next chapters, we propose methods for MIL classification for each case, each posing their own specific challenges. The second chapter proposes a

general purpose method for bag-level classification under the standard MIL assumption. The method addresses several challenges such as the noisy features, multimodal distributions and low witness rates. The next chapter proposes a method for bag classification under the collective assumption for personality assessment in speech signals. This problem is challenging because the label space for instances is different than for bags. Finally, in Chapter 4, we address instance-level classification problems in an active learning framework. Instance-level classification poses specific challenges because the misclassification cost of instances is different than for bag-level classification and cannot be used directly in the optimization. Moreover, these problems are often associated with severe class imbalance. Next, a more detailed overview of each chapter is presented.

The first chapter contains an overview of MIL from the point of view of the important characteristics that make MIL problems unique. The MIL assumptions and related tasks are discussed first. Then, we present a recapitulation of the general literature about MIL problems and methods. After, we proceed with explaining what makes MIL different from other types of learning. Among several other subjects, the distinction between instance-level and bag-level classification is thoroughly discussed, as well as the possible types of relations between instances, the effect of label ambiguity and data distributions. Relevant methods for each characteristic are surveyed. Next, we review MIL formulation for different applications and relate these applications to the problem characteristics. Finally, we conduct experiments where we compare 16 reference methods under various conditions and draw several conclusions. The paper ends on a discussion containing recommendation for experimental protocols, complexity and future directions. This part of the thesis is at its second round of revision for publication in Elsevier's "Pattern Recognition" (Carbonneau *et al.*, 2016a).

The second chapter extends a method presented in the previous conference publication (see Annex I). The method is called Random Subspace for Witness Identification (RSWI). In the

MIL literature, a positive instance is often called a witness. The method is used to classify instance individually given a collection of labeled bags. In (Garcia-Garcia & Williamson, 2011), a distinction is made between inductive and transductive learning scenario. In the inductive learning scenario, the goal is to train a learner to make inference on new data. This is the classical classification scenario: a classifier learns a decision function using training data in the hope it will generalize well on test data. In the transductive scenario, one aims to discover the structure of data given a finite data set. This corresponds to the classical clustering scenario where one learns the structure of a data set. In that case, there is no test data, the goal is thus to obtain an understanding of the data structure. In this paper, RSWI is used in the transductive scenario: the method is used to classify instance individually given a collection of labeled bags. In this chapter a similar method is used to build a bag-level classifier in an inductive learning scenario. The method is called Random Subspace for Instance Selection (RSIS). In that case, the method determines the likelihood of each instance to be a witness. These likelihoods are used to sample training sets which are used to train a pool of classifiers. Each classifier in the pool is an instance classifier. To perform bag-level classification, predictions for each instance of the bag are combined. The method exhibits high robustness to noisy features and performs well with various types of positive and negative distributions. Furthermore, the method is robust to the proportion of positive instances per positive bag hereafter called low witness rates (WR). This chapter was published in Elsevier's Pattern Recognition (Carbonneau *et al.*, 2016e).

The third chapter presents a MIL method proposed to infer speaker personality from speech segments. This application in challenging because it is not possible to pinpoint which part of the signal is responsible for class assignation. In fact, personality is a complex concept and it is unlikely that a single instance defines the personality of a speaker over an entire speech segment. On the contrary, personality manifests in a series interrelated cues. This means that the label space for instances is different from the label space for bags. Therefore, the collective MIL assumption must be employed instead of the standard MIL assumption. Moreover, the

relations between instances which must be considered because they convey important information. The method proposed in the paper is akin to a BoW, which embeds the content of a bag in a code vector and trains a classifier on these code vectors. While presenting a MIL method, the paper focuses on how to represent speech signals of various lengths in a meaningful way. First, a temporal signal is transformed into a spectrogram from which patches are extracted. Then, the speech signal is represented as a collection of spectrogram patches. In the MIL vocabulary, signals are bags and patches are instances. A dictionary of concepts is learned from all training patches using a sparse coding formulation. All patches are encoded as a composition of the learned concepts in the dictionary. These instances are sum-aggregated to obtain the code vector representing the whole bag. The method obtains state-of-the-art results on real-world data with a highly reduced complexity when compared to commonly used approaches in the field. This chapter is in its second round of revision for publication in IEEE transactions on Affective Computing.

In the fourth chapter, active learning methods are proposed in the context MIL instance classification. The particular structure of MIL problems makes SI active learners suboptimal in this context. We propose to tackle the problem from two different perspectives sometimes referred to as the two faces of active learning (Dasgupta, 2011). The first method, aggregated informativeness (AGIN), identifies the bags containing the most informative instances based on their proximity to the classifier decision boundary. The second method, cluster-based aggregative sampling (C-BAS), discovers the cluster structure of the data. It characterizes each cluster based on how much is known about the cluster composition and the level of conflict between bag and instance labels. Bags are selected based on the membership of instances to promising clusters. The performance of both methods is examined in inductive and transductive learning scenarios. This chapter has been submitted to IEEE Transactions on Neural Networks and Learning Systems in October 2017.

CHAPTER 1

LITERATURE REVIEW: MULTIPLE INSTANCE LEARNING: A SURVEY OF PROBLEM CHARACTERISTICS AND APPLICATIONS

Marc-André Carbonneau[1,2], Veronika Cheplygina[3], Eric Granger[1], Ghyslain Gagnon[2]

[1] Laboratory for Imagery, Vision and Artificial Intelligence,
École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] Communications and Microelectronic Integration Laboratory,
École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[3] Medical Image Analysis group, Eindhoven University of Technology,
De Rondom 70, 5612 AP Eindhoven, Holland

Article under a second round of revision in « Elsevier's Pattern Recognition ». Initially submitted in January 2017.

Abstract

Multiple instance learning (MIL) is a form of weakly supervised learning where training instances are arranged in sets, called bags, and a label is provided for the entire bag. This formulation is gaining interest because it naturally fits various problems and allows to leverage weakly labeled data. Consequently, it has been used in diverse application fields such as computer vision and document classification. However, learning from bags raises important challenges that are unique to MIL. This paper provides a comprehensive survey of the characteristics which define and differentiate the types of MIL problems. Until now, these problem characteristics have not been formally identified and described. As a result, the variations in performance of MIL algorithms from one data set to another are difficult to explain. In this paper, MIL problem characteristics are grouped into four broad categories: the composition of the bags, the types of data distribution, the ambiguity of instance labels, and the task to be performed. Methods specialized to address each category are reviewed. Then, the extent to which these characteristics manifest themselves in key MIL application areas are described. Finally, experiments are conducted to compare the performance of 16 state-of-the-art MIL methods on

selected problem characteristics. This paper provides insight on how the problem characteristics affect MIL algorithms, recommendations for future benchmarking and promising avenues for research. Code is available on-line at https://github.com/macarbonneau/MILSurvey.

1.1 Introduction

Multiple instance learning (MIL) deals with training data arranged in sets, called bags. Supervision is provided only for entire sets, and the individual label of the instances contained in the bags are not provided. This problem formulation has attracted much attention from the research community, especially in the recent years, where the amount of data needed to address large problems has increased exponentially. Large quantities of data necessitate a growing labeling effort.

Weakly supervised methods, such as MIL, can alleviate this burden since weak supervision is generally obtained more efficiently. For example, object detectors can be trained with images collected from the web using their associated tags as weak supervision, instead of locally-annotated data sets (Hoffman *et al.*, 2015; Wu *et al.*, 2015b). Computer-aided diagnosis algorithms can be trained with medical images for which only patient diagnoses are available instead of costly local annotations provided by an expert. Moreover, there are several types of problems that can naturally be formulated as MIL problems. For example, in the drug activity prediction problem (Dietterich *et al.*, 1997), the objective is to predict if a molecule induces a given effect. A molecule can take many conformations which can either produce, or not, a desired effect. Observing the effect of individual conformations is unfeasible. Therefore, molecules must be observed as a group of conformations, hence use the MIL formulation. Because of these attractive properties, MIL has been increasingly used in many other application fields over the last 20 years, such as image and video classification (Chen *et al.*, 2006; Rahmani & Goldman, 2006; Andrews *et al.*, 2002; Zhang *et al.*, 2002; Phan *et al.*, 2015; Cinbis *et al.*, 2016), document classification (Zhou *et al.*, 2009; Bunescu & Mooney, 2007a) and sound classification (Briggs *et al.*, 2012).

Several comparative studies and meta-analyses have been published to better understand MIL (Zhou, 2004; Babenko, 2008; Amores, 2013; Doran & Ray, 2014a; Alpaydın *et al.*, 2015; Ray & Craven, 2005; Cheplygina *et al.*, 2015d; Vanwinckelen *et al.*, 2015; Alpaydın *et al.*, 2015; Cheplygina *et al.*, 2015b; Cheplygina & Tax, 2015; Foulds & Frank, 2010). All these papers observe that the performance of MIL algorithms depends on the characteristics of the problem. While some of these characteristics have been partially analyzed in the literature (Zhou *et al.*, 2009; Bunescu & Mooney, 2007a; Li & Sminchisescu, 2010; Han *et al.*, 2010), a formal definition of key MIL problem characteristics has yet to be described.

A limited understanding of such fundamental problem characteristics affects the advancement of MIL research in many ways. Experimental results can be difficult to interpret, proposed algorithms are evaluated on inappropriate benchmark data sets, and results on synthetic data often do not generalize to real-world data. Moreover, characteristics associated with MIL problems have been addressed under different names. For example, the scenario where the number of positive instances in a bag is low was referred to as either sparse bags (Yan *et al.*, 2016; Bunescu & Mooney, 2007b) or low witness rate (Li & Sminchisescu, 2010; Li *et al.*, 2013). It is thus important for future research to formally identify and analyze what defines and differentiates MIL problems.

This paper provides a comprehensive survey of the characteristics inherent to MIL problems, and investigates their impact on the performance of MIL algorithms. These problem characteristics are all related to unique features of MIL: the ambiguity of instance labels and the grouping of data in bags. We propose to organize problem characteristics in four broad categories: *Prediction level*, *Bag composition*, *Label ambiguity* and *Data distribution*.

Each characteristic raises different challenges. When instances are grouped in bags, predictions can be performed at two levels: bags-level or instance-level (Cheplygina *et al.*, 2015d). These two tasks have different misclassification costs therefore algorithms are often better suited for only one of them (Vanwinckelen *et al.*, 2015; Alpaydın *et al.*, 2015) (A more detailed discussion is presented in Section 1.4.1). Bag composition, such as the proportion of instances from

each class and the relation between instances, also affects the performance of MIL methods. The source of ambiguity on instance labels is another important factor to consider. This ambiguity can be related to label noise as well as to instances not belonging to clearly defined classes (Foulds & Frank, 2010). Finally, the shape of positive and negative distributions affects MIL algorithms depending on their assumptions about the data.

As additional contributions, this paper reviews state-of-the-art methods which can address challenges of each problem characteristic. It also examines several applications of MIL, and in each case, identifies their main characteristics and challenges. For example, in computer vision, instances can be spatially related, but this relationship does not exist in most bioinformatics applications. Finally, experiments show the effects of selected problem characteristics – the instance classification task, witness rate, negative class modeling and label noise – with 16 representative MIL algorithms. This is the first time that algorithms are compared on the bag and instance classification tasks in the light of these specific challenges. Our findings indicate that these problem characteristics have a considerable impact on the performance of all MIL methods, and that each method is affected differently. Therefore, problem characterization cannot be ignored when proposing new MIL methods and conducting comparative experiments. Finally, this paper provides novel insights and direction to orient future research in this field from the problem characteristics point-of-view.

The rest of this paper is organized as follows. The next section describes MIL assumptions and the different learning tasks that can be performed using the MIL framework. Section 1.3 reviews previous surveys and general MIL studies. Section 1.4 and 1.5 identify and analyze the key problem characteristics and applications, respectively. Experiments are presented in Section 4.4, followed by a discussion in Section 1.7.

1.2 Multiple Instance Learning

1.2.1 Assumptions

In this paper, we consider two broad assumptions: the standard and the collective assumption. For a more detailed review on the subject, the reader is referred to (Foulds & Frank, 2010).

The *standard MIL assumption* states that all negative bags contain only negative instances, and that positive bags contain at least one positive instance. These positive instances are named witnesses in many papers and this designation is used in this survey. Let X be a bag defined as a set of feature vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$. Each instance (i.e. feature vector) $\mathbf{x}_i$ in feature space $\mathcal{X}$ can be mapped to a class by some process $f : \mathcal{X} \to \{0, 1\}$, where the negative and positive classes correspond 0 and 1 respectively. The bag classifier $g(X)$ is defined by:

$$g(X) = \begin{cases} 1, & \text{if } \exists \mathbf{x} \in X : f(\mathbf{x}) = 1; \\ 0, & \text{otherwise,} \end{cases} \tag{1.1}$$

This is the working assumption of many of the early methods (Dietterich *et al.*, 1997; Andrews *et al.*, 2002; Maron & Lozano-Pérez, 1998), as well as recent ones (Carbonneau *et al.*, 2016e; Xiao *et al.*, 2016). To correctly classify bags under the standard assumption, it is not necessary to identify all witnesses as long as at least one is found in each positive bag. A more detailed discussion will be presented in Section 1.4.1.

The standard MIL assumption can be relaxed to address problems where positive bags cannot be identified by a single instance, but by the distribution, interaction or accumulation of the instances it contains. Here, instances in a bag are no longer independent and bag classifiers can take many forms. We will give three representative examples in this section.

In some problems, several positive instances are necessary to assign a positive label to a bag. For example, in traffic jam detection from images of a road, a car would be a positive instance.

However, an image containing a few cars is not positive because it takes many cars to create a traffic jam. In this case a bag classifier can be given by:

$$g(X) = \begin{cases} 1, & \text{if } \theta \leq \sum_{\mathbf{x} \in X} f(\mathbf{x}); \\ 0, & \text{otherwise,} \end{cases} \tag{1.2}$$

where θ is the minimal number of witnesses in positive bags.

A more general case for the collective assumption is when bags are defined positive by instances belonging to more than one concept. Foulds and Frank (Foulds & Frank, 2010) give a simple and representative example of this assumption by classifying images of desert, sea and beach. Images of deserts will contain sand segments, while images of the sea contain water segments. However, images of beaches must contain both types of segments. To correctly classify beach images, the model must verify the presence of both types of witnesses, and thus, methods working under the standard MIL assumption would fail in this case. Some methods assign instances to a set of defined concepts ($\mathcal{C}$), and some of these concepts belong to the positive class ($\mathcal{C}^+ \subset \mathcal{C}$). In that case, the bag classifier $g(X)$ is defined by:

$$g(X) = \begin{cases} 1, & \text{if } \forall c \in \mathcal{C}^+ : \theta_c \leq \sum_{\mathbf{x} \in X} f_c(\mathbf{x}); \\ 0, & \text{otherwise,} \end{cases} \tag{1.3}$$

where $f_c(\mathbf{x})$ is a process that outputs 1 if $\mathbf{x}$ belongs to concept c and θ_c is the number of instances belonging to c required to observe a positive bag. There are different levels of generality for multiple concepts assumptions of this type (Weidmann *et al.*, 2003). Alternatively, bag can be seen as distributions of instances. In (Doran, 2015), the bag space $\mathcal{B}$ is defined as the set of all probability distributions on the instance space ($\mathcal{P}(\mathcal{X})$). Each bag X is a probability distribution over instances $P(\mathbf{x}|X)$. In that case a bag classifier is a process that maps a probability distribution to a label: $g(X) : \mathcal{P}(\mathcal{X}) \rightarrow \{0,1\}$.

In this survey, the *collective assumption* designates all assumptions in which more than one instance are needed to identify a positive bag.

1.2.2 Tasks

Classification: Classification can be performed at two levels: bag and instance. Bag classification is the most common task for MIL algorithms. It consists in assigning a class label to a set of instances. The individual instance labels are not necessarily important depending on the type of algorithm and assumption. Instance classification is different from bag classification because while training is performed using data arranged in sets, the objective is to classify instance individually. As pointed out in (Carbonneau *et al.*, 2016d), the loss functions for the two tasks are different (see Section 1.4.1). When the goal is bag classification, misclassifying an instance does not necessarily affect the loss at bag-level. For example, in a positive bag, few true negative instances can be erroneously classified as positive and the bag label will remain unchanged. Thus, the structure of the problem, such as the number of instances in bags, plays an important role in the loss function (Vanwinckelen *et al.*, 2015). As a result, the performance of an algorithm for bag classification is not representative of the performance obtained for instance classification. Moreover, many methods proposed for bag classification (e.g. (Zhang & Goldman, 2001; Zhou & Zhang, 2007)) do not reason in instance space, and thus, often cannot perform instance classification.

MIL classification is not limited to assigning a single label to instances or bags. Assigning multiple labels to bags is particularly relevant considering that they can contain instances representing different concepts. This idea has been the object of several publications (Zha *et al.*, 2008; **?**). Multi-label classification is subject to the same problem characteristics as single label classification, thus no distinction will be made between the two in the rest of this paper.

Regression: MIL regression task consists in assigning a real value to a bag (or an instance) instead of a class label. The problem has been approached in different ways. Some methods assign the bag label based on a single instance. This instance may be the closest to a target concept (Dooly *et al.*, 2003), or the best fit in a regression model (Ray & Page, 2001). Other methods work under the collective assumption and use the average or a weighted combination of the instances to represent bags as a single feature vector (Wang *et al.*, 2008b; Wagstaff & Lane,

2007; Pappas & Popescu-Belis, 2014). Alternatively, on can simply replace a bag-level classifier by a regressor (EL-Manzalawy *et al.*, 2011).

Ranking: Some methods have been proposed to rank bags or instances instead of assigning a class label or a score. The problem differs from regression because the goal is not to obtain an exact real valued label, but to compare the magnitude of scores to perform sorting. Ranking can be performed at the bag-level (Bergeron *et al.*, 2012) or at the instance-level (Hu *et al.*, 2008).

Clustering: This task consists in finding clusters or a structure among a set of unlabeled bags. The literature on the subject is limited. In some cases, clustering is performed in bag space using standard algorithms and set-based distance measures (e.g. k-Medoids and the Hausdorff distance (Zhang & Zhou, 2009)). Alternatively, clustering can be performed at the instance-level. For example, in (Zhang *et al.*, 2011a), the algorithm identifies the most relevant instance of each bag, and performs maximum margin clustering on these instances.

Most of the discussion in the remainder of the paper will be articulated around classification, as it is the most studied task. However, challenges and conclusions related to problem characteristics are also applicable to the other tasks.

1.3 Studies on MIL

Because many problems can be formulated as MIL, there is a plethora of MIL algorithms in the literature. However, there is only a handful of general MIL studies and surveys. This section summarizes and interprets the broad conclusions from these general MIL papers.

The first survey on MIL is a technical report written in 2004 (Zhou, 2004). It describes several MIL algorithms, some applications and discusses learnability under the MIL framework. In 2008, Babenko published a report (Babenko, 2008) containing an updated survey of the main families of MIL methods, and distinguished two types of ambiguity in MIL problems. The first type is polymorphism ambiguity, in which each instance is a distinct entity or a distinct version

of an entity (e.g. conformations of a molecule). The second is part-whole ambiguity in which all instances are parts of the same object (e.g. segments of an image). In a more recent survey (Amores, 2013), Amores proposed a taxonomy in which MIL methods are divided in three broad categories following the representation space. Methods operating in the instance-space are grouped together, and the methods operating in bag-space are divided in two categories based on whether a bag embedding is performed or not. Several experiments were performed to compare bag classification accuracy in four application fields. Bag-space methods performed better in terms of bag classification accuracy, however, performance depends on the data and the distance function or the embedding method. Recently, a book on MIL has been published (Herrera *et al.*, 2016a). It discusses most of the tasks of Section 1.2.2 along with associated methods, as well as data reduction and imbalanced data. Finally, Quellec et al. (Quellec *et al.*, 2017) wrote a survey on MIL for medical imaging applications, for which MIL is a particularly attractive solution. They review how problems are formulated in this field of applications and analyze results from various experiments. They conclude that, while being more convenient, MIL outperforms single instance learning because it can pick up on subtle global visual cues that cannot be properly segmented and used as single instances to train a classifier.

Some papers study specific topics of MIL. For instance, Foulds and Frank reviewed the assumptions (Foulds & Frank, 2010) made by MIL algorithms. They stated that these assumptions influence how algorithms perform on different types of data sets. They found that algorithms working under the collective assumption also perform well with data sets corresponding to the standard MIL assumption, such as the Musk data set (Dietterich *et al.*, 1997). Sabato and Tishby (Sabato & Tishby, 2012) analyzed the of sample complexity in MIL, and they found that the statistical performance of MIL is only mildly dependent on the number of instances per bag. In (Cheplygina & Tax, 2015) the similarities between MIL benchmark data sets were studied. The data sets were represented in two ways: by meta-features describing numbers of bags, instances and so forth, and by features based on performances of MIL algorithms. Both representations were embedded in a 2-D space and found to be dissimilar to each other. In other words, data sets often considered similar due to the application or size of data did

not behave similarly, which suggest that some unobserved properties influence MIL algorithm performances.

Some papers compare MIL to other learning settings to better understand when to use MIL. Ray and Craven (Ray & Craven, 2005) compared the performance of MIL methods against supervised methods on MIL problems. They found that in many cases, supervised methods yield the most competitive results. They also noted that, while some methods systematically dominate others, the performance of the algorithms was application-dependent. In (Cheplygina *et al.*, 2015d), the relationship between MIL and settings such as group-based classification and set classification is explored. They state that MIL is applicable in two scenarios: the classification of bags and the classification of instances. Recently, these differences were rigorously investigated (Vanwinckelen *et al.*, 2015). It was shown analytically and experimentally that the correlation between classification performance at bag and instance level is relatively weak. Experiments showed that depending on the data set, the best algorithm for bag classification provides average, or even the worst performance for instance classification. They too observed that different MIL algorithms perform differently given the nature of the data.

The classification of instances can be a task in itself, but can also be an intermediate step toward bag classification for instance-space methods (Amores, 2013). Alpaydin et al. (Alpaydın *et al.*, 2015) compared instance-space and bag-space classifiers on synthetic and real-world data. They concluded that for datasets with few bags, it is preferable to use an instance-space classifier. They also state, as in (Amores, 2013), that if the instances provide partial information about the bag labels, it is preferable to use bag-space representation. In (Cheplygina *et al.*, 2015b), Cheplygina et al. explored the stability of the instance labels assigned by MIL algorithms. They found that algorithms yielding best bag classification performance were not the algorithms providing the most consistent instance labels. Carbonneau et al. (Carbonneau *et al.*, 2016c) studied the ability to identify witnesses (positive instances) of several MIL methods. They found that depending on the nature of the data, some algorithms perform well while others would have difficulty learning.

Finally, some papers focus on specific classes of algorithms and applications. Doran and Ray (Doran & Ray, 2014a) analyzed and compared several SVM-based MIL methods. They found that some methods perform better for instance classification than for bag classification, or vice-versa, depending on the method properties. Wei and Zhou (Wei & Zhou, 2016) compared methods for generating bags of instances from images. They found that sampling instances densely leads to a higher accuracy than sampling instances at interest points or after segmentation. This agrees with other bag-of-words (BoW) empirical comparisons (Nowak *et al.*, 2006; Wang *et al.*, 2009). They also found that methods using the collective assumption performed better for image classification. Vankatesan et al. (Venkatesan *et al.*, 2015) showed that simple lazy-learning techniques could be applied to some MIL problems to obtain results comparable to state-of-the-art techniques. Kandemir and Hamprecht (Kandemir & Hamprecht, 2015) compared several MIL algorithms in two computer-aided diagnosis (CAD) applications. They found that modeling intra-bag similarities was a good strategy for bag classification in this context.

The main conclusions of these studies are summarized as follows:

- The performance of MIL algorithms depends on several properties of the data set (Amores, 2013; Ray & Craven, 2005; Vanwinckelen *et al.*, 2015; Alpaydın *et al.*, 2015; Cheplygina & Tax, 2015; Carbonneau *et al.*, 2016c);
- When it is necessary to model combinations of instances to infer bag labels, bag-space and embedding methods perform better (Amores, 2013; Alpaydın *et al.*, 2015; Quellec *et al.*, 2017; Wei & Zhou, 2016);
- The best bag-level classifier is rarely the best instance-level classifier, and vice versa (Doran & Ray, 2014a; Vanwinckelen *et al.*, 2015);
- When the number of bags is low, it is preferable to use an instance-based method (Alpaydın *et al.*, 2015);

- Some MIL problems can also be solved effectively using standard supervised methods (Ray & Craven, 2005);
- Performance of MIL is only mildly dependent on the number of instances per bag (Sabato & Tishby, 2012);
- Similarity between the instances of a same bag affect classification performance (Kandemir & Hamprecht, 2015).

All of these conclusions are related to one or more characteristics that are unique to MIL problems. **Identifying these characteristics and gaining a better understanding of their impact on MIL algorithms is an important step towards the advancement of MIL research.** This survey paper mainly focuses on these characteristics and their implications for methods and applications. For a more general survey on MIL methods, we refer the interested reader to (Amores, 2013).

1.4 Characteristics of MIL Problems

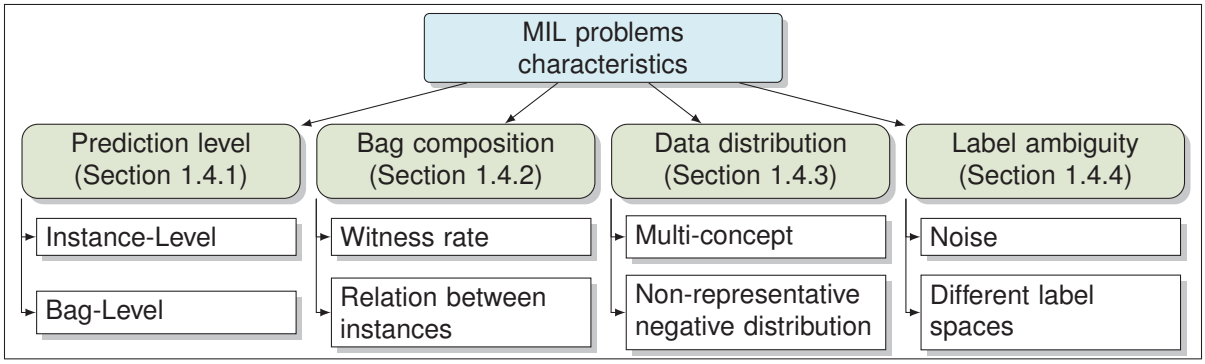


Figure 1.1 Characteristics inherent to MIL problems

We identified four broad categories of key characteristics associated with MIL problems which directly impact on the behavior of MIL algorithms: *prediction level*, *bag composition*, *data distributions* and *label ambiguity* (as shown in Fig. 1.1). Each characteristic poses different challenges which must be addressed specifically.

In the remainder of this section, each of these characteristics will be discussed in more detail, along with representative specialized methods proposed in the literature to address them.

1.4.1 Prediction: Instance-level vs. Bag-level

In some applications, like object localization in images, the objective is not to classify bags, but to classify individual instances. In that case, problems are formulated with the implicit assumption that instances can be labeled as positive or negative. Following the notation of Section 1.2.1, for instance classification, the task is to learn $f(\mathbf{x})$ rather than $g(\mathbf{x})$. These two tasks are related in the sense that a perfect instance classifier $f^*(\mathbf{x})$ would result in a perfect bag classifier under the standard MIL assumption:

$$g^*(X) = \begin{cases} 1, & \text{if } \exists \mathbf{x} \in X : f^*(\mathbf{x}) = 1; \\ 0, & \text{otherwise,} \end{cases} \tag{1.4}$$

Inversely, a perfect bag classifier $g^*(X)$ achieves perfect instance classification since an instance can be viewed as a singleton bag, $S = \{\mathbf{x}\}$:

$$g^*(S) = f^*(\mathbf{x}). \tag{1.5}$$

In practice, none of these optimal classifiers are likely to be trained. More importantly, the relation between optimal classifiers for a given finite data set is no longer reciprocal. A perfect instance classifier still leads to an optimal bag classifier but the inverse is not true. For example, suppose all instances of a MIL data set are sampled from either one of two positive concepts (C_1 and C_2) or from a negative concept (C_-). In addition, all positive bags contain positive instances from both positive concepts and from the negative concept: $X^+ = \{\mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2, \mathbf{x}_3 \in C_-\}$. All negative bags contain instances sampled from the negative concept: $X^- = \{\mathbf{x}_1 \in C_-, \mathbf{x}_2 \in$

$C_-, ..., \mathbf{x}_N \in C_-\}$. The following classifier achieves perfect bag classification:

$$\hat{g}^*(X) = \begin{cases} 1, & \text{if } \exists \mathbf{x} \in X : \hat{f}(\mathbf{x}) = 1; \\ 0, & \text{otherwise,} \end{cases} \tag{1.6}$$

where

$$\hat{f}(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in C_1; \\ 0, & \text{otherwise.} \end{cases} \tag{1.7}$$

While $\hat{g}^*(X)$ would correctly classify all bags in the data set, $\hat{f}(\mathbf{x})$ would misclassify half of the positive instances.

In MIL, training an instance classifier is non-trivial because instance labels are unavailable. This is why many methods use bag classification accuracy (e.g. APR (Dietterich *et al.*, 1997), MI-SVM (Andrews *et al.*, 2002), MIL-Boost (Babenko *et al.*, 2008), EM-DD (Zhang & Goldman, 2001), MILD (Li & Yeung, 2010)) as a surrogate optimization objective to train an instance classifier in the hope that bag-level accuracy will be representative of instance-level accuracy. However, as will be discussed next, there are key differences in the cost function of the two tasks. These differences explain why the bag-level accuracy of a method does not reflect its accuracy at instance-level (Doran & Ray, 2014a; Vanwinckelen *et al.*, 2015). It was shown in analytic and empirical investigations (Vanwinckelen *et al.*, 2015) that the relationship between the accuracy at the two levels depends of the number of instances in bags, the class imbalance and the accuracy of the instance classifier. It follows that algorithms designed for bag classification are not optimal for instance classification.

Here we explain the difference between the instance misclassification cost for both classification levels. Under the standard MIL assumption, as soon as a witness is identified in a bag, it is labeled as positive and all other instance labels can be ignored. In that case, false positives (FP) and false negatives (FN) have no impact on the bag classification accuracy, but still count as classification errors at the instance level. In addition, when considering negative bags, a single FP causes a bag to be misclassified. This means that if 1% of the instances in each negative

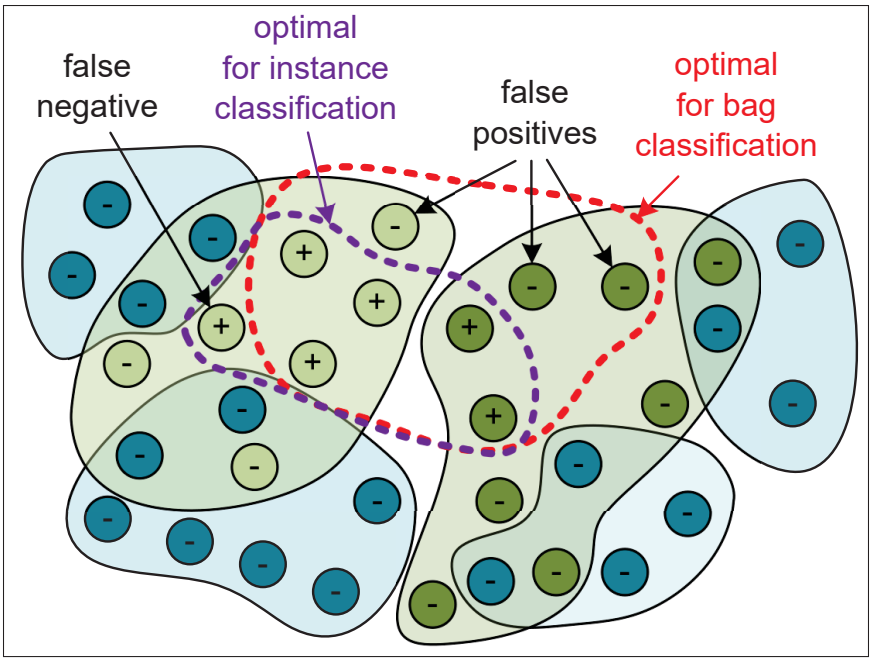


Figure 1.2 Illustration of two decisions boundaries on a fictive problem. While only the purple boundary correctly classifies all instances, both them achieve perfect bag classification. This is because, in that case, false positive and false negative instances do not impact on bag labels

bag were misclassified, the accuracy on negative bags would be 0%, although the accuracy on negative instances would be 99%. This is illustrated in Fig. 1.2. The green ensembles represent positive bags, while negative bags correspond to blue ensembles. Each instance is identified with its true class. In this figure, both decision boundaries (dotted lines) are optimal for bag classification because they include at least one instance from all positive bags, while excluding all instances from negative bags. However, only one of the two boundaries achieves perfect instance classification (purple).

The vast majority of methods in the literature address the bag classification problem. These methods have been extensively surveyed in the past thus we refer the interested reader to (Zhou, 2004; Babenko, 2008; Amores, 2013). A large proportion of the methods proposed for instance classification use a measure bag classification accuracy to train an instance classifier. The predictions for all instances from a bag are aggregated, generally using the max function (or a

differentiable approximation), and the loss is computed with respect to the bag label. This idea has been used to train a Boosting classifier in (Babenko *et al.*, 2011c; Viola *et al.*, 2006) and other types of model such as logistic regression (Ray & Craven, 2005) and deep neural networks (Wu *et al.*, 2015b). The aforementioned methods were proposed for instance classification but are not different in spirit from most bag classification methods reasoning in the instance space like APR (Dietterich *et al.*, 1997), EM-DD (Zhang & Goldman, 2001), MI-OptimalBall (Auer & Ortner, 2004), MI-SVM (Andrews *et al.*, 2002) and SDB-MIL (Xiao *et al.*, 2016). These methods classify instances individually before predicting bag labels which means they can directly be used for instance-level classification.

As explained above, using bag classification accuracy as a surrogate optimization objective is suboptimal. This is why it has been proposed to consider negative and positive bags separately in the classifier loss function (Jia & Zhang, 2008). The accuracy on positive bags is taken at bag level, but for negative bags, all instances are treated individually. This optimization criterion was proposed to adjust the decision threshold of bag classifiers for instance classification and improve their accuracy in (Carbonneau *et al.*, 2016d). In (Yang *et al.*, 2006), a different weight is assigned to FP and FN during the optimization of an SVM. Virtually any bag-level classifier can classify instances if they are seen as singleton bags. This is the rationale behind Citation-kNN-ROI (Zhou *et al.*, 2005b) which does not perform well in practice (see Section 1.6.2). MILES (Chen *et al.*, 2006) is a bag classification method based on prototype distance embedding and SVM that can be used for instance classification. The method computes the contribution of each instance to the bag label assignation based on its distance to selected prototypes. Instances in positive bags for which the contribution is above a given threshold are identified as witnesses.

Some methods try to uncover the true label of the instances to train an instance classifier. One of the most well-known methods is mi-SVM (Andrews *et al.*, 2002). After instances labels have been initialized, an SVM classifier is trained and used to update the label assignation. These two steps are performed iteratively until the label assignation remains unchanged. The resulting SVM classifier is used to predict the label of test instances. MissSVM (Zhou & Xu,

2007) views the problem as semi-supervised learning where the instance in positive bags are unlabeled. The algorithm is similar to mi-SVM except that the constraint that every positive bags contain a positive instance is enforced. KI-SVM (Li *et al.*, 2009) uses a multiple kernel approach in which a kernel encodes possible label assignations in the SVM constraints. In this method, it is assumed that there are the same number of positive instances in all positive bags. MILD (Li & Yeung, 2010) discovers a set of *true positive* instances. The probability that an instance is positive depends on the bag labels in its vicinity defined by a Gaussian kernel. The discovered true positive instances are used to train an SVM classifier. A similar idea is proposed in RSIS-EoSVM (Carbonneau *et al.*, 2016e) where instances are projected in random subspaces and vicinity depends on cluster assignations. In that case, label assignation is probabilistic. Several training sets are sampled based on these probabilistic assignations to train an ensemble of SVM classifiers.

1.4.2 Bag Composition

Witness Rate

The witness rate (WR) is the proportion of positive instances in positive bags. When the WR is very high, positive bags contain only a few negative instances. In that case, the label of the instances can be assumed be the same as the label of their bag. The problem then reverts to a supervised problem with one-sided noise which can be solved in a regular supervised framework (Blum & Kalai, 1998). However, in some applications, WR can be arbitrarily small and hinder the performance of many algorithms. For example, in methods like Diverse Density (DD) (Maron & Lozano-Pérez, 1998), Citation-kNN (Zhang & Goldman, 2001) and APR (Dietterich *et al.*, 1997) instances are considered to have the same label as their bag. When the WR is low, this is no longer reasonable and leads to lower performances. Methods which analyze instance distributions in bags (Amores, 2010; Doran & Ray, 2014b; Wei *et al.*, 2014) may also have problems dealing with low WR because distribution in positive and negative bags become similar. Also, some methods represent bags by the average of the instances they

contain, like NSK-SVM (Gärtner *et al.*, 2002), or by considering their contribution to the bag label equally (Xu & Frank, 2004). With very low WRs, the few positive instances have a limited effect after the pooling process. Finally, in instance classification problems, lower WRs mean serious class imbalance problems, which leads to bad performance for many methods.

Several authors studied low WR problems in recent years. For example, sparse transductive MIL (stMIL) (Bunescu & Mooney, 2007b) is an SVM formulation similar to NSK-SVM (Gärtner *et al.*, 2002). However, to better deal with low WR bags, the optimization constraints of the SVM are modified to be satisfied when at least one witness is found in positive bags. This method performs well at low WR but is less efficient when it is higher. Sparse balanced MIL (sbMIL) (Bunescu & Mooney, 2007b) incorporates an estimation of the WR as a parameter in the optimization objective to solve this problem. WR estimation has also been successfully used in low WR problems by ALP-SVM (Gehler & Chapelle, 2007), SVR-SVM (Li & Sminchisescu, 2010) and the γ-rule (Li *et al.*, 2013). One drawback of using the WR as a parameter is that the WR is assumed to be constant across all bags. Other methods, like CR-MILBoost (Ali & Saenko, 2014) and RSIS (Carbonneau *et al.*, 2016e), estimate the probability that each instance is positive before training an ensemble of classifiers. During training, the classifiers give more importance to the instances that are more likely to be witnesses. In miGraph (Zhou *et al.*, 2009), similar instances in a bag are grouped in cliques. The importance of each instance is inversely proportional to the size of its clique. Assuming positive and negative instances belong to different cliques, the WR has little impact. In miDoc (Yan *et al.*, 2016), a graph represents the entire MIL problem, where bags are compared based on the connecting edges. Experiments show that the method performs well on very low WR problems.

Relations Between Instances

Most existing MIL methods assume, often not explicitly, that positive and negative instances are sampled independently from a positive and a negative distribution. However, this is rarely the case with real-world data. In many applications, the i.i.d. assumption is violated because structure or correlations exist between instances and bags (Zhou *et al.*, 2009; Zhang *et al.*,

2011b). We make a distinction between three types of relation: intra-bag similarities, instance co-occurrences and structure.

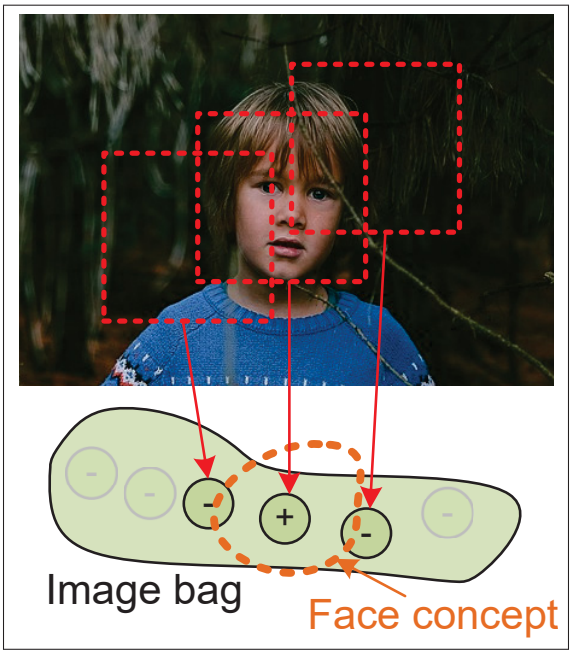


Figure 1.3 Illustration of intra-bag similarity between instances: The patches are overlapping, and thus, share similarities with each other

Intra-Bag Similarities: In some problems, instances belonging to the same bag share similarities that instances from other bags do not. For instance, in the drug activity prediction problem (Dietterich *et al.*, 1997), each bag contains many conformations of the same molecule. It is likely that instances of the same molecule are similar to some extent, while being different from other molecules (Zhou, 2004). One must ensure that the MIL algorithm learns to differentiate active from non-active conformations, instead of learning to classify molecules. In image-related applications, it is likely that all segments share some similarities related the capture conditions (e.g. illumination, noise, etc.). Alternatively, similarities between instances of a same bag may be related to the instance generation process. For example, some methods use densely extracted patches which overlap (Figure 1.3). Since they share a certain number

of pixels, they are likely to be correlated. Also, the background of a picture could be split in different segments which can be very similar (see Figure 1.4).

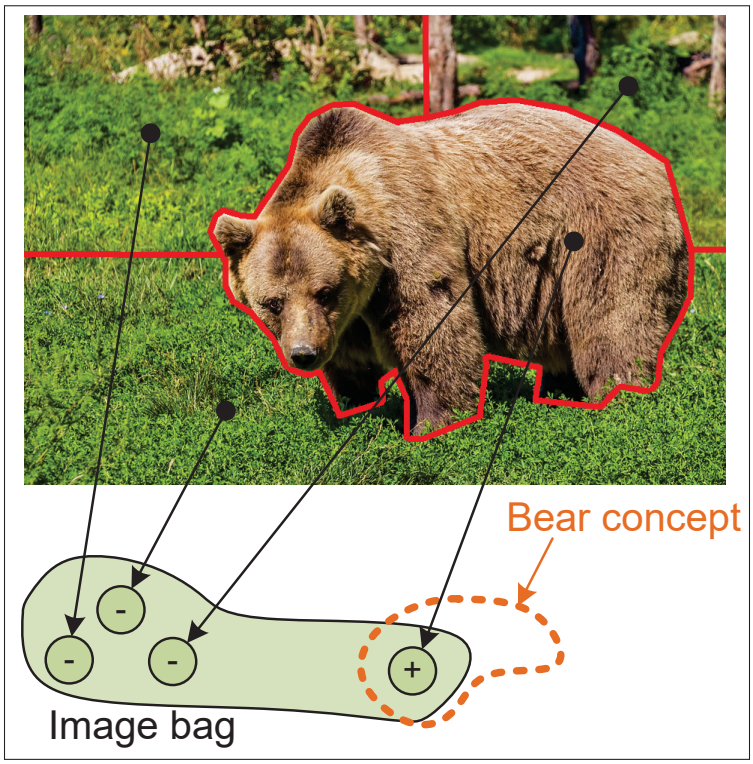


Figure 1.4 Example of co-occurrence and similarity between instances: Three segments contain grass and forest and are therefore very similar. Moreover, since this is an image of a bear, the background is more likely to be nature than a nuclear central control room

Intra-bag similarities raise some challenges during learning. For instance, transductive algorithms (e.g. mi-SVM (Andrews *et al.*, 2002)) may not be able to infer instance labels if the nature of negative instances from positive and negative bags differ (Ray & Craven, 2005).

Very few methods were proposed explicitly to address this problem. To deal with similar instances, miGraph (Zhou *et al.*, 2009) builds a graph per bag and groups similar instances together to adjust their relative importance based on the group size. CCE (Zhou & Zhang, 2007) performs a clustering of the instance space. Bags are represented by a binary vector

in which each bit corresponds to a cluster. A bit is set to one if at least one instance in the bag has been assigned to the corresponding cluster. A similar approach is used in (Wu *et al.*, 2014b) except bits are associated a pool of subgraphs patterns mined from the data set. Because features are binary, many instances can be assigned to the same cluster and the representation remains unaffected, which provides robustness to intra-bag similarity.

Instances are similar if they are close to each other in the metric space used by the classifier. Depending on the type of data, similarity or dissimilarity can be measured using different distance measures such as Euclidean (Chai *et al.*, 2014a), cosine (Yan *et al.*, 2016) or χ^2 (Laptev *et al.*, 2008). A good way to mitigate problems related to intra-class similarity is to define a new instance space in which distance are more related to class than bag membership. This new space can be obtained by selecting features which truly discriminate between class (instead of bags) or by learning a representation in which class discriminant information is enhanced. In most cases, the new reduced instance space maximizes the distance between negative instances and the most positive instance of each positive bag. For example, Relief-MI (Zafra *et al.*, 2012) is an adaptation of the Relief (Kononenko, 1994) feature selection algorithm for MIL. For random bags, it identifies the nearest neighbors from each class under different versions of the Hausdorff distance. Then, it assigns a score to each feature based on the distance difference between the neighbor of the same class and the others under this feature. The most discriminant features are selected and the others are discarded. Other feature selection algorithms have been adapted for MIL in a similar fashion (Zafra & Ventura, 2010; Zafra *et al.*, 2013). In B-M3IFW (Chai *et al.*, 2014a), a positive bag is represented by its most positive instance to form a pool of positive prototypes. Feature weights are obtained by maximizing a margin defined as the difference between two terms: the distance between positive prototypes and negative instances and the distance between positive prototypes the mean of positive prototypes.

Several methods include built-in feature selection or weighting mechanisms. For instance, APR (Dietterich *et al.*, 1997) searches for a subset of features in which a hyper-rectangle encompassing at least one instance from all positive bags while keeping negative instances

outside. MIRVM (Raykar *et al.*, 2008) performs classification and feature selection at the same time in a Bayesian learning framework. It uses MILR (Ray & Craven, 2005) and perform optimal feature selection with the type-II maximum likelihood method. Diverse Density (Maron & Lozano-Pérez, 1998; Zhang & Goldman, 2001) scales the importance of each feature to define the optimal region encompassing the positive concept in the instance space. This scaling has also been used in (Zhang & Zhou, 2004) to increase the performance of a BP-MIP (Zhou & Zhang, 2002).

Finally, feature learning methods project instances in a space of reduced dimensionality where class discrimination at bag level is enforced. Usually this means maximizing the distance between negative instances and the most positive instance of each positive bag in the projection space. This can be achieved using MIL adaptation of discriminant analysis or other linear projection method (Ping *et al.*, 2010; Kim & Choi, 2010; Chai *et al.*, 2014b; Sun *et al.*, 2010) where bag classification accuracy is maximized.

Instance Co-occurrence: Instances co-occur in bags when they share a semantic relation. This type of correlation happens when the subject of a picture is more likely to be seen in some environment than in another, or when some objects are often found together (e.g. knife and fork). For example, the bear of Figure 1.4 is more likely to be found in nature than in a nightclub. Thus, the observation of nature segments might help to decide if the image contains a cocktail or a bear (Kang *et al.*, 2006). In (Cheplygina *et al.*, 2015c), it is shown that different birds are often heard in the same audio fragment, so a "negative" bird song could help to correctly classify the bird of interest. In these examples, co-occurrence represents an opportunity for better accuracy, however, in some cases it is a necessary condition for successful classification. Consider the example given by Foulds and Frank (Foulds & Frank, 2010) where one must classify sea, desert and beach images. Both desert and beach images can contain sand instances, while water instances can be found in sea and beach images. However, both instances must co-occur in a beach image. Most methods working under the collective assumption (Foulds & Frank, 2010) naturally leverage co-occurrence. Many of these methods, like BoW (Amores, 2010; Csurka *et al.*, 2004), miFV (Wei *et al.*, 2014), FAMER (Ping *et al.*,

2011) or PPMM (Wang *et al.*, 2008a) represent bags as instance distributions which indirectly account for co-occurrence. This has also been directly modeled in a tensor model (Qi *et al.*, 2007) and in a multi-label framework (Zha *et al.*, 2008).

While useful to classify bags, in instance classification problems, the co-occurrence of instances may confuse the learner. If a given positive instance often co-occurs with a given negative instance, the algorithm is more likely to consider the negative instance as positive, which in this context would lead to a higher false positive rate (FPR).

Instance and Bag Structure: In some problems, there exists an underlying structure between instances in bags or even between bags (Zhang *et al.*, 2011b). Structure is more complex than simple co-occurrence in the sense that instances follow a certain order, or are related in a meaningful way. Capturing this structure may lead to better classification performance (Zhou *et al.*, 2009; Laptev *et al.*, 2008; Ryoo & Aggarwal, 2009). The structure may be spatial, temporal, relational or even causal. For example, when a bag represents a video sequence, all frames or patches are temporally and spatially ordered. For example, it is difficult to differentiate between a person taking or leaving a package without taking this temporal order into account. Alternatively, in web mining tasks (Zhang *et al.*, 2011b) where websites are bags and pages linked by the websites are instances, there exists a semantic relation between two bags representing websites linked together.

Graph models were proposed to better capture the relations between the different entities in non-i.i.d. MIL problems. Structure can be exploited at many levels: graphs can be used to model the relations between bags, instances or both (Yan *et al.*, 2016; Zhang *et al.*, 2011b). Graphs enforce that related objects belong to the same class. Alternatively, (Mcgovern & Jensen, 2003) represents bags as graphs capturing diverse relationships between objects. The objects are shared across all bags and all possible sub-graphs of the bag graph correspond to instances. In (Wu *et al.*, 2014b, 2015a), complex objects such as web pages and scientific papers are represented as a collection of graphs. Discriminative subgraph patterns are mined to create a dictionary. Graph collections are represented by binary feature vectors in which each bit cor-

responds a subgraph in the dictionary. A bit is set to 1 if the corresponding subgraph is part of the collection. In (Bi & Liang, 2007), spatial structure in the image is captured by a similarity matrix and a neighborhood consistency constraint is enforced in a 1-norm SVM formulation.

Temporal and spatial structure between instances can be modeled in different ways. In BoW models, this can be achieved by dividing the images (Grauman & Darrell, 2005; Lazebnik *et al.*, 2006) or videos (Laptev *et al.*, 2008) into different spatial and/or temporal zones. Each zone is characterized individually, and the final representation is the concatenation of every zone feature vectors. For audio and video, sub-sequences of instances have been analyzed using traditional sequence modeling tools such as conditional random fields (CRF) (Tax *et al.*, 2010) and hidden Markov model (HMM) (Guan *et al.*, 2016). Spatial dependency in images have also been modeled in with CRF in (Zha *et al.*, 2008; Warrell & Torr, 2011).

1.4.3 Data Distributions

Many methods make implicit assumptions on the shape of the distributions, or on how well the negative distribution is represented by the training set. In this section, the challenges associated with the nature of the overall data distribution is studied.

Multimodal Distributions of Positive Instances

Some MIL algorithms work under the assumption that the positive instances are located in a single cluster or region in feature space. This is the case for several early methods like APR (Dietterich *et al.*, 1997), which searches for a hyper-rectangle that maximizes the inclusion of instances from positive bags while excluding instances from negative bags. Diverse Density (DD) (Maron & Lozano-Pérez, 1998) methods follow a similar idea. These methods locate the point in feature space closest to instances in positive bags, but far from instances in negative bags. This point is considered to be the positive concept. Some more recent methods follow the single cluster assumption. CKMIL (Li *et al.*, 2014) locates the most positive instance in each bag based on its proximity to a single positive cluster center. In (Xiao *et al.*, 2016), the

classifier is a sphere encompassing at least one positive instance from each positive bag while excluding instances from negative bags. The method in (Tax *et al.*, 2010) employs a similar strategy.

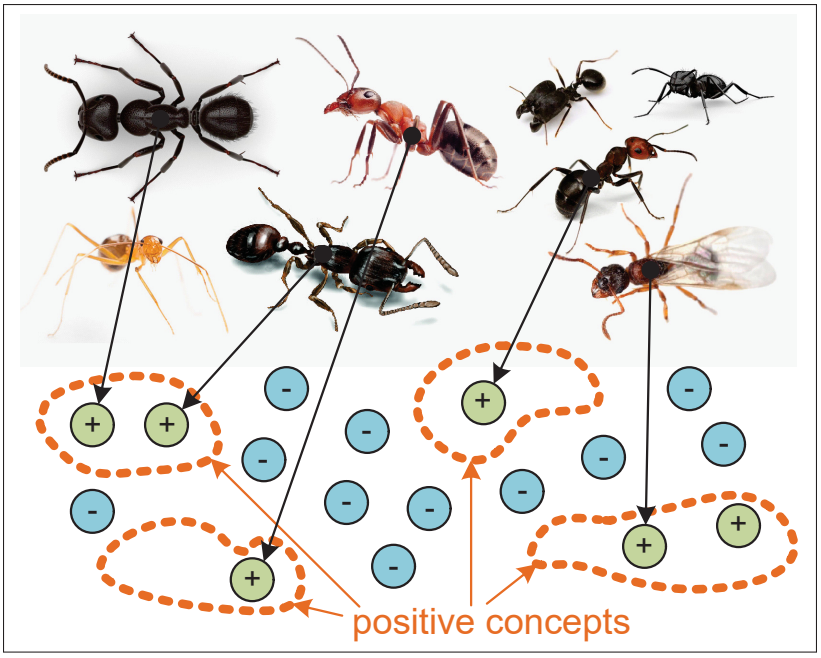


Figure 1.5 For the same concept *ants*, there can be many data clusters (modes) in feature space corresponding to different poses, colors and castes

The single cluster assumption is reasonable in some applications such as molecule classification, but problematic in many other contexts. In image classification, the target concept may correspond to many clusters. For example, Fig. 1.5, shows several pictures of ants. Ants can be black, red or yellow, they can have wings and different body shapes depending on the species and castes. Their appearance also changes depending on the point-of-view. It is unlikely that a compact location in feature space encompasses all of these variations.

Many MIL methods can learn multimodal positive concepts, however, only few representative approaches will be mentioned due to space constraints. First, non-parametric methods based on distance between bags like Citation-kNN(Wang & Zucker, 2000) and MInD (Cheplygina *et al.*, 2015c) naturally deal with all shapes of distributions. Simple non-parametric methods

often lead to competitive results in MIL problems (Venkatesan *et al.*, 2015). Methods using distances to a set of prototypes as bag representation, like DD-SVM (Chen & Wang, 2004) and MILES (Chen *et al.*, 2006), can model many positive clusters, because each different cluster can be represented by a different prototype. Instance-space SVM-based methods like mi-SVM (Andrews *et al.*, 2002) can deal with disjoint regions of positive instances using a kernel. Also, methods modeling instance distributions in bags such as vocabulary-based (Amores, 2010) methods naturally deal with data sets containing multiple concepts/modes. The mixture-model in (Wang *et al.*, 2012) naturally represents different positive clusters. In (Carbonneau *et al.*, 2016e) instances are grouped in clusters and the composition of the clusters are analyzed to compute the probability that instances are positive.

Non-Representative Negative Distribution

In (Doran, 2015), it is stated that learnability of instance concept requires that the distribution in test is identical to the training distribution. This is true for positive concepts, however, in some applications, the training data cannot entirely represent the negative instance distribution. For instance, provided sufficient training data, it is reasonable to expect that an algorithm learns a meaningful representation that captures the visual concept of a human person. However, since humans can be found in many different environments, ranging from jungle to spaceships, it is almost impossible to entirely model the negative class distribution. In contrast, in some applications like tumor identification in radiography, healthy tissue regions compose the negative class. These tissues possess a limited appearance range that can be modeled using a finite number of samples.

Several methods model only the positive class, and thus are well-equipped to deal with different negative distributions in test. In most cases, these methods search for a region encompassing the positive concept. In APR (Dietterich *et al.*, 1997) the region is a hyper-rectangle, while in many others it is one, or a collection of, hyper-spheres/-ellipses (Maron & Lozano-Pérez, 1998; Xiao *et al.*, 2016; Zhang & Goldman, 2001; Tax & Duin, 2008). These methods perform classification based on the distance to a point (concept) or a region in feature space. Everything

that is far enough from the point, or outside the positive region, is considered negative. Therefore, the shape of the negative distribution is unimportant. A similar argument can be made for some non-parametric methods such as Citation-kNN (Wang & Zucker, 2000). These methods use the distance to positive instances, instead of positive concepts, and thus, offer the same advantage. Alternatively, the MIL problem can be seen as a one-class problem, where positive instances are the target class. Consequently, several methods using one-class SVM have been proposed (Zhang *et al.*, 2005; Wu & Chung, 2009; Wang *et al.*, 2016).

Experiments in Section 1.6.4 compare reference MIL algorithms in contexts where the negative distribution is different in training and in test.

1.4.4 Label Ambiguity

Label ambiguity is inherent to weak supervision. In MIL, this ambiguity can take different forms depending on the assumption under which the problem is formulated. Under the standard MIL assumption, there is no ambiguity on instance labels in negative bags. In that case, MIL can be viewed as a special kind of semi-supervised problem (Zhou & Xu, 2007) where the labeled portion of the data belongs to only one class and the instance are structured in sets with label constraints. Under more relaxed MIL assumptions, there are supplementary sources of ambiguity such as noise on labels and instance labels different from bag labels.

Label Noise

Some MIL algorithms, especially those working under the standard MIL assumption, rely heavily on the correctness of bag labels. For instance, it was shown in (Venkatesan *et al.*, 2015) that DD is not tolerant to noise in the sense that a single negative instance in the neighborhood of the positive concept can hinder performances. A similar argument was made for APR (Li & Yeung, 2010) for which a negative bag mislabeled as positive, would lead to a high FPR.

In practice, there are many situations where positive instances may be found in negative bags. There are situations where labeling errors occur, but sometimes labeling noise is inherent to the data. For example, in computer vision applications, it is difficult to guarantee that negative images contain no positive patches: An image showing a house may contain flowers, but is unlikely to be annotated as a flower image (Li & Vasconcelos, 2015). Similar problems may arise in text classification, where a paragraph contains an analogy and thus, uses words from another subject.

Methods working under the collective assumption can naturally deal with label noise. Positive instances found in negative bags have less impact, because these methods do not assign label solely based on the presence of a single positive instance. The methods representing bags as distributions (Amores, 2010; Doran & Ray, 2014b; Rubner *et al.*, 2000) can naturally deal with noisy instances because a single positive instance does not significantly change the distribution of a negative bag. Methods summarizing bags by averaging the instances like NSK-kernel (Gärtner *et al.*, 2002) also provide robustness to noise in a similar manner. Another strategy to deal with noise is to count the number of positive instances in bags, and establish a threshold for positive classification. This is referred as the threshold-based MI Assumption in (Foulds & Frank, 2010). The method proposed (Li & Vasconcelos, 2015) uses both the thresholding and the averaging strategies. The instances of a bag are ranked from most positive to less positive, and the bags are represented by the mean of the top-ranking instances and the mean of the bottom ranking instances. The averaging operation mitigates the effects of positive instance in negative bags. In (Erdem & Erdem, 2011), robustness to label noise is obtained by using dominant sets to perform clustering and select relevant instance prototype in a bag-embedding algorithm similar to MILES (Chen *et al.*, 2006).

Experiments in Section 1.6.5 compare the robustness to label noise of the reference methods.

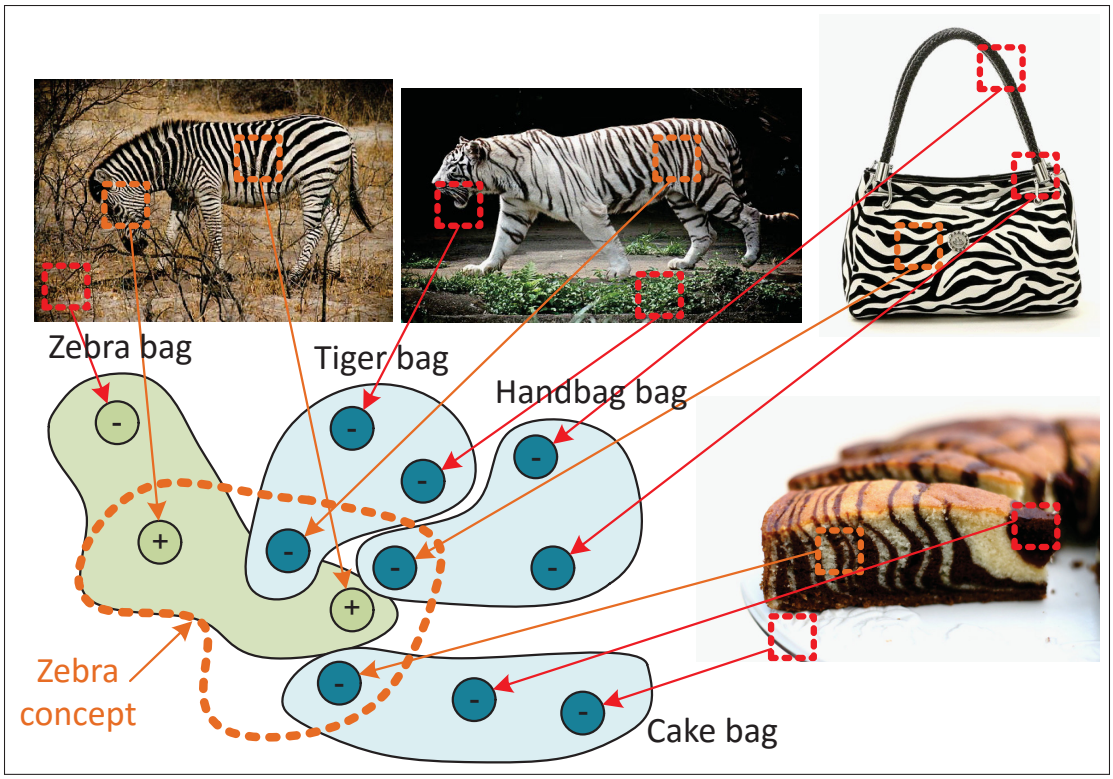


Figure 1.6 This is an example of instances with ambiguous labels. *Zebra* is the target concept and instances relating to this concept should fall in the region delimited by the dotted line. However, negative images can also contain instances falling inside the zebra concept region

Different Label Spaces

There are MIL problems in which the label space for instances is different from the label space for bags. In some cases, these spaces will correspond to different granularity levels. For example, a bag labeled as a car will contain instances labeled as wheel, windshield, headlights, etc. In other cases, instances labels might not have clear semantic meanings. Fig. 1.6 shows an example where the positive concept is zebra (represented by the region encompassed by the orange dotted line). This region contains several types of patches that can be extracted from a zebra picture. However, it is possible to extract patches from negative images that fall into this positive region. In this example, some patches extracted from the image of a white tiger, a purse and a marble cake fall into the zebra concept region. In that case the patches do not have semantic meaning easily understandable by humans.

When instances cannot be assigned to a specific class, methods operating under the standard MIL assumption, which must identify positive instances, are inadequate. Therefore, in those cases, using the collective assumption is necessary. Vocabulary-based methods (Amores, 2010) are particularly well adapted for this situation. They associate instances to words (e.g. prototypes or clusters) discovered from the instance distribution. Bags are represented by distributions over these words. Similarly, methods using embedding based on distance from selected prototype instance, such as MILES (Chen *et al.*, 2006) and MILIS (Fu *et al.*, 2011), can also deal with this type of problem.

All the characteristics presented in this section define a variety of MIL problem, which each must be addressed differently. The next section relates these characteristics to the prominent application fields of MIL.

1.5 Applications

MIL represents a powerful approach that is used in different application fields mostly (1) to solve problems where instances are naturally arranged in sets and (2) to leverage weakly annotated data.

This section surveys the main application fields of MIL. Each field is examined with respect to their different problem characteristics of Section 1.4 (summarized in Table 1.1).

1.5.1 Biology and Chemistry

The problems in biology and chemistry can often be naturally formulated as MIL problems because of the inability to observe individual instance classes. For instance, in the molecule classification task presented in the seminal paper by Dietterich et al. (Dietterich *et al.*, 1997), the objective is to predict if a molecule will be binding to a musk receptor. Each molecule can take many conformations, with different binding strengths. It is not possible to observe the binding strength of a single conformation, but it is possible to observe it for groups of conformations, hence the MIL problem formulation.

Table 1.1 Typical problem characteristics associated with MIL in literature for different application fields (Legend: ✓ likely to have a moderate impact, ✓✓ likely to have a large impact on performance)

	Problem Characteristics									
Application Fields	Instance classification	Real-valued outputs	Low witness rate	Intra-bag similarities	Instance co-occurence	Structure in bags	Multimodal positive distribution	Non-modelable negative distribution	Label noise	Different label spaces
Drug activity prediction	✓	✓		✓✓			✓	✓		
DNA Protein identification	✓✓	✓	✓	✓✓		✓✓	✓	✓		
Binding sites identification	✓✓	✓		✓✓			✓	✓		
Image Retrieval			✓	✓	✓✓	✓✓	✓✓	✓✓	✓	✓✓
Object localization in image	✓✓		✓	✓	✓	✓	✓✓	✓✓	✓✓	✓
Object localization in video	✓✓		✓	✓	✓	✓✓	✓✓	✓✓	✓✓	✓
Computer aided diagnosis	✓	✓	✓	✓	✓		✓		✓✓	✓
Text classification	✓		✓		✓✓		✓✓	✓	✓	✓
Web mining	✓		✓	✓	✓	✓	✓	✓		✓
Sound classification	✓			✓	✓	✓✓	✓	✓	✓	
Activity recognition	✓				✓	✓✓	✓	✓	✓	✓

Since then, MIL has found use in many drug design and biological applications. Usually, the approach is similar to Dietterich's: complex chemical or biological entities (compounds, molecules, genes, etc.) are modeled as bags. These entities are composed of parts or regions that can induce an effect of interest. The goal is to classify unknown bags and sometimes to identify witness to better understand underlying mechanisms of the biological or chemical phenomenon. MIL has been used, among others, to predict a drug's bioavailability (Bergeron *et al.*, 2012), predict the binding affinity of peptides to major histocompatibility complex molecules (EL-Manzalawy *et al.*, 2011), discover binding sites governing gene expression (Bandyopadhyay *et al.*, 2015; Palachanis, 2014) and predict gene functions (Eksi *et al.*, 2013).

The problems presented in this section are of various natures, but there are some characteristics that apply to a majority of them. For example, in most cases, the bags represent many arrangements or viewpoints of the same entity (e.g. drug, genes, etc.), which translate into

high intra-bag similarities. Also, many applications call for quantification, using ranking or regression (Dooly *et al.*, 2003) (e.g. quantifying the binding strength of a molecule), which is more difficult and less documented than classification. Some characteristics apply only to a type of application. Some objects like DNA sequences produce structured bags, while the many conformations of the same molecule do not. Finally, some problems require the identification of entities responsible for an effect (e.g. drug binding). This calls for methods with instance classification capabilities.

1.5.2 Computer Vision

MIL is used in computer vision for two main reasons: to characterize complex visual concepts using sets of different sub-concepts, and to learn from weakly annotated data. The next subsections describe how MIL is used for content-based image retrieval (CBIR) and object localization. MIL is gaining momentum in the medical imaging community, and a subsection will also be devoted to this application field.

Content Based Image Retrieval

CBIR is probably the single most popular application of MIL. The list of publications addressing this problem is long (Chen *et al.*, 2006; Rahmani & Goldman, 2006; Andrews *et al.*, 2002; Zhang *et al.*, 2002, 2005; Vijayanarasimhan & Grauman, 2008; Maron & Ratan, 1998; Leistner *et al.*, 2010; Song *et al.*, 2013). The task in CBIR is to categorize images based on the objects/concepts they contain. The exact localization of the objects is not important, which means it is primarily a bag classification problem. Typically, images are partitioned into smaller parts or segments, which are then described by feature vectors. Each segment corresponds to an instance, while the whole image corresponds to a bag. Images can be partitioned in many ways, which are compared in (Wei & Zhou, 2016). For example, the image can be partitioned using a regular grid (Maron & Ratan, 1998), key-points (Csurka *et al.*, 2004) or semantic regions (Yang *et al.*, 2006; Chen & Wang, 2004). In the latter case, the images are divided using

state-of-the-art segmentation algorithms. This limits instance ambiguity since segments tend to contain only one object.

Visual data poses several challenges to MIL algorithms mainly because images are a good example of non-i.i.d. data. For one, some objects are more likely to co-occur in the same picture than others (e.g. bird and sky). Methods leveraging these co-occurrences tend to be more successful. Also, a bag can contain many similar instances, especially if the instances are obtained using dense grid sampling. Methods using segmentation algorithms are less subject to this problem since segments tend to correspond to single objects. Sometimes, the image is composed of several concepts, which means methods working under the collective MIL assumption perform better. Moreover, working with images often means working with large intra-class variability. The same object can, for instance, appear considerably different depending on the points of view. Many types of object also come is a variety of shapes and colors. This means it is unlikely that a unimodal distribution adequately represents the entire class. Finally, backgrounds can vary considerably, making it difficult to learn a negative distribution that models every possible background object.

Object Localization and Segmentation

In MIL, the localization of objects in images (or videos) means learning from bags to classify instances. Typically, MIL is used to train visual object recognition systems on weakly labeled image data sets. In other words, labels are assigned to entire images based on the objects they contain. The objects do not have to be in the foreground, and an image may contain multiple objects. In contrast, in strongly supervised applications, bounding boxes indicating the location of each object are provided along with object labels. In other cases, pixel-wise annotations are provided instead. These bounding boxes, or pixel annotations, are often manually specified, and thus, necessitate considerable human effort. The computer vision community turned to MIL to leverage the large quantity of weakly annotated images found on the Internet to build object detectors. The weak supervision can come from description sentences (Xu *et al.*, 2016; Karpathy & Fei-Fei, 2015; Fang *et al.*, 2015), web search engine results (Zhu *et al.*, 2015), tags

associated with similar images and words found on web pages associated with the images (Wu *et al.*, 2015b).

In several methods for object localization, bags are composed of many candidate bounding boxes corresponding to instances (Hoffman *et al.*, 2015; Babenko *et al.*, 2008; Song *et al.*, 2014; Babenko *et al.*, 2011b; Sapienza *et al.*, 2014). The best bounding box to encompass the target object is assumed to be the most positive instance in the bag. Efforts were dedicated to localize objects and segment them at pixel-level using traditional segmentation algorithms such as Constraint Parametric Min-Cuts (Müller & Behnke, 2012), JSEG (Zha *et al.*, 2008) or Multi-scale combinatorial grouping (Hariharan *et al.*, 2014). Alternatively, segmentation can be achieved by casting each pixel of the image as an instance (Vezhnevets & Buhmann, 2010).

Instance classification has also been applied in videos. It has been used to recognize complex events such as "attempting a board trick" or "birthday party" (Phan *et al.*, 2015; Lai *et al.*, 2014). Several concepts compose these complex events. Evidence of these concepts sometimes lasts only for a short time, and can be difficult to observe in the total amount of information presented in the video. To deal with this problem, video sequences are divided in shorter sequences (instances) that are later classified individually. This problem formulation is also used in (Wang *et al.*, 2011) to recognize scenes that are inappropriate for children. Also in videos, MIL methods were proposed to perform object tracking (Babenko *et al.*, 2011c; Zhang & Song, 2013; Lu *et al.*, 2011). For example, in (Babenko *et al.*, 2011c) a classifier is trained online to recognize and track an object of interest in a frame sequence. The tracker proposes candidate windows which compose a bag and are used to train the MIL classifier.

It can be difficult to manually select a finite set of classes to represent every object found in a set of images. Thus, it was proposed to perform the object localization alongside class discovery (Zhu *et al.*, 2015). The method is akin to multiple instance clustering methods (Zhang & Zhou, 2009; Zhang *et al.*, 2011a), but generates bags using a saliency detector, which remove background objects from positive bags to achieve higher cluster purity. A method based on multiple

instance clustering was also proposed to discover a set of actons (sub-actions) from videos to create a mid-level representation of actions (Zhu *et al.*, 2013).

Object localization is susceptible to the same challenges as CBIR: instances in images are correlated, exhibit high similarity and spatial (and temporal for videos) structures exist in the bags. The objects can be deformable, have various appearances and be observed from different viewpoints. Therefore, a single concept is often represented by a multimodal distribution, and the negative distribution cannot be entirely captured by a training set. However, object localization is different from CBIR because it is an instance classification problem, which means that many bag-level algorithms are inapplicable. Several authors have also noted that in this context, MIL algorithms are sensitive to initialization (Cinbis *et al.*, 2016; Song *et al.*, 2014).

Computer Aided Diagnosis and Detection

MIL is gaining popularity in medical applications. Weak labels, such as the overall diagnosis of a subject, are typically easier to obtain than strong labels, such as outlines of abnormalities in a medical scan. The MIL framework is appropriate in this situation given that patients have both abnormal and healthy regions in their medical scan, while healthy subjects have only healthy regions. The diseases and image modalities used are very diverse; applications include classification of cancer in histopathology images (Xu *et al.*, 2014), diabetes in retinal images (Quellec *et al.*, 2012), dementia in brain MR (Tong *et al.*, 2014), tuberculosis in X-ray images (Melendez *et al.*, 2015a), classification of a chronic lung disease in CT (Cheplygina *et al.*, 2014) and others.

Like in other general computer vision tasks, there are two main goals in these applications: diagnosis (i.e. predicting labels for subjects), and detection or segmentation (i.e. predicting labels for a part of a scan). These parts can be pixels or voxels (3D pixel), an image patch or a region of interest. Different applications pursue one or both goals, and have different reasons for doing so.

When the focus is on classifying bags, MIL classifiers benefit from using information about co-occurrence and structure of instances. For example, in (Melendez *et al.*, 2015a), a MIL classifier trained only with X-ray images labeled as healthy or as containing tuberculosis, outperforms its supervised version, trained on outlines of tuberculosis lesions. Similar results are observed on the task of classification of chronic obstructive pulmonary disease (COPD) from chest computed tomography images (Cheplygina *et al.*, 2014).

Literature that is focused on classifying instances is somewhat less common, which may be a consequence of the lack of instance-labeled datasets. However, the lack of instance labels is what is often the motivation for using MIL in the first place, which means instance-level evaluation is necessary if these classifiers are to be translated into clinical practice. Some papers do not perform instance-level evaluation because the classifier does not provide such output (Tong *et al.*, 2014), but state that this would be a useful extension of the method in the future. Others provide instance labels but do not have access to ground truth, thus resorting to more qualitative evaluation. For example, (Cheplygina *et al.*, 2014) examines whether the instances classified as "most positive" by the classifier have similar intensity distributions to what is already known in the literature. Finally, when instance-level labels are available, the classifier can be evaluated quantitatively and/or qualitatively. Quantitative evaluation is performed in (Kandemir & Hamprecht, 2015; Quellec *et al.*, 2012; Melendez *et al.*, 2015a). In addition, the output of the classifier can be displayed in the image, which is an interpretable way of visualizing the results. In (Melendez *et al.*, 2015a), the mi-SVM classifier provides local real-valued tuberculosis abnormality scores for each pixel in the image, which are then visualized as a heatmap on top of the X-ray image.

CAD shares many key challenges with other less constrained computer vision tasks. Depending on the sampling – which can be done on a dense grid (Kandemir & Hamprecht, 2015; Melendez *et al.*, 2015a), randomly (Cheplygina *et al.*, 2014) or according to constraints (Tong *et al.*, 2014) – the instances can display varying degrees of similarity. In many pathologies, abnormalities are likely to include different subtypes, which have different appearance resulting in multimodal concept distributions. Moreover, differences between patients, such as age, sex

and weight, as well as differences in acquisition of the images can also lead to large intra-class variability. On the other hand, the negative distribution (healthy tissue) is more constrained than in computer vision applications. This means that it is a reasonable to attempt to capture and model the negative distribution, which is very difficult in unconstrained image recognition problems. Another particularity of CAD problems is that they are naturally suitable to have real-valued outputs, because diseases can have different stages, although this is often not considered when off-the-shelf algorithms are applied. For example, the chronic lung disease COPD has 4 different stages, but (Cheplygina *et al.*, 2014) treats them all as the positive class. During evaluation, the mild stage is most often misclassified as healthy. (Tong *et al.*, 2014) considers binary classification tasks out of four possible classes (healthy, two types of mild cognitive impairment, and Alzheimer's), while these could be considered as a continuous scale. Lastly, CAD can be formulated as an instance and a bag classification task.

1.5.3 Document Classification and Web Mining

Considering the Bag-of-Words (BoW) model is a MIL model working under the collective assumption, document classification is one of the earliest (1954) applications of MIL (Harris, 1954). BoW represents texts as frequency histograms quantifying the occurrence of each word in the text. In this context, texts and web pages are multi-part entities that require MIL classification framework.

Texts often contain several topics and are easily modeled as bags. Text classification problems can be formulated as MIL at different levels. At the lowest level, instances are words like in the BoW model. Alternatively, instances can be sentences (Pappas & Popescu-Belis, 2014; Zhang *et al.*, 2008), passages (Andrews *et al.*, 2002; Zhang *et al.*, 2013) or paragraphs (Ray & Craven, 2005). In (Andrews *et al.*, 2002), bags are text documents, which are divided in overlapping passages corresponding to instances. The passages are represented by a binary vector in which each element is a medical term. The task is to categorize the texts. In (Settles *et al.*, 2008), instances are short posts from different newsgroups. A bag is a collection of posts and the task is to determine if a group of posts contains a reference to a subject of interest. In (Ray & Craven,

2005), the task consists of identifying texts that contain a passage which links a protein to a particular component, process or function. In this case, paragraphs are instances while entire texts are bags. The paragraphs are represented by a BoW alongside distances from the protein names and key terms. In (Jorgensen *et al.*, 2008), the content of emails is analyzed to detect spam. A common approach to elude spam filters is to include words that are not associated with spam in the message. Representing emails as bags of passages proved to be an efficient way to deal with these attacks. In (Pappas & Popescu-Belis, 2014; Zhang *et al.*, 2008; Kotzias *et al.*, 2014, 2015), MIL was used to infer the sentiment expressed in individual sentences based on the labels provided for entire user reviews. MIL has also been used to discover relations between named entities (Bunescu & Mooney, 2007a). In this case, bags are collections of sentences containing two words that may or may not express a target relation (e.g. "Rick Astley" lives in "Montreal"). If the two words are related in the specified way, some of the sentences in the bag will express this relation. If that is not the case, none of the sentences will indicate the relation, hence the MIL formulation.

Web pages can also be naturally modeled using the MIL framework. Just like texts, web pages often contain many topics. For instance, a news channel website contains several articles on a diversity of subjects. MIL has been used for web index-page recommendations based on a user browsing history (Zhou *et al.*, 2005a; Zafra *et al.*, 2007). A web index page contains links, titles and sometimes short description of web pages. In this context, a web index page is a bag, and the linked web pages are the instances. Following the standard MIL assumption, it is hypothesized that if a web index page is marked as favorite, the user is interested in a least one of the pages linked to it. Web pages are represented by the set of the most frequent terms they contain. In contextual web advertisement, advertisers prefer to avoid certain pages containing sensitive content like war or pornography. In (Zhang *et al.*, 2008), a MIL classifier assesses sections of web pages to identify suitable web pages for advertisement.

Text data poses particular challenges for MIL. Most of the time instances are non-i.i.d. Words may have different meanings depending on the context and thus, co-occurrence is important in this type of application. While BoW methods are successful to some degree, structure is

an important component of sentences which convey important semantic information. Often only small passages or specific words indicate the class of the document, which means WR can be quite low. Depending on the task and the formulation of the problem, bag and instance classification can be performed. In addition, text classification can present an additional difficulty compared to other applications. When texts are represented by word frequency features (e.g. BoW) the data is very sparse and high-dimensional (Andrews *et al.*, 2002). This type of data is often difficult to handle by classifiers using Euclidean-like distance measures. These distributions are highly multimodal and it is difficult to adequately represent the distribution of negative data.

1.5.4 Other Applications

The MIL formulation has found its way to various other application fields. In this section, we present some less common applications for MIL along with their respective formulation.

Reinforcement learning (RL) shares some similarities with MIL. In both cases, only a weak supervision is provided for the instances. In RL, a reward, the weak supervision, is assigned to a state/action pair. The reward obtained for the state/action pair is not necessarily directly related to it, but might be related to preceding actions and states. Consider a RL agent learning how to play chess. The agent obtains a reward (or punishment) only at the end of the game. In other words, a label is given for a collection (bag) of action/state pairs (instances). This correspondence has motivated the use of MIL to accelerate RL by the discovery of sub-goals in a task (Mcgovern & Jensen, 2003). These sub-goals are, in fact, the positive instances in the successful episodes. The main challenge for RL tasks is to consider the structure in bags and the label noise since good actions can be found in bad sequences.

Just like for images, some sound classification tasks can be cast as MIL. In (Mandel & Ellis, 2008), the objective is to automatically determine the genre of musical excerpts. In training, labels are provided for entire albums or artists, but not for each excerpt. The bags are collection of excerpts from the same artist or album. It is possible to find different genres of music on

the same album or from the same artist, therefore the bags may contain positive and negative instances. In (Briggs *et al.*, 2012), MIL is used to identify bird songs in recordings made by an unattended microphone in the wild. Sound sequences contain several types of birds and other noises. The objective is to identify each birdsong individually while training only on weakly labeled sound files.

Some methods represent audio signals as spectrograms and use image recognition techniques to perform recognition (Lyon, 2010). This idea has been used for bird song recognition (Ruiz-Muñoz *et al.*, 2015) with histograms of gradients. In (Carbonneau *et al.*, 2016b), personality traits are inferred from speech signals represented as spectrograms in a BoW framework. In that case, entire speech signals are bags and small parts of the spectrogram are instances. The BoW framework has been used in a similar fashion in (Kumar & Raj, 2016), however, in that case instances are cepstrum feature vectors representing 1 second-long audio segments. Audio classification poses different challenges depending on how sounds are represented. For example, when a sound signal is represented as a time series, capturing structure is important. However, in a BoW framework, the co-occurrence of different markers will be more important. In many cases, the background noise related to capture conditions leads to high intra-bag similarity.

Time series are found in several applications other than audio classification. For instance, in (Guan *et al.*, 2016; Stikic *et al.*, 2011) MIL is used to recognize human activities from wearable body sensors. The weak supervision comes from the users stating which activities were performed in a given time period. Typically, activities do not span across entire periods and each period may contain different activities. In this setup, instances are sub-periods, while the entire periods are bags. A similar model is used for the prediction of hard drive failure (Murray *et al.*, 2005). In this case, time series are a set of measurements on hard drives taken at regular intervals. The goal is to predict when a product is about to fail. Time series imply structure in bags that should not be ignored.

In (Manandhar *et al.*, 2012; Karem & Frigui, 2011), MIL classifiers detect buried landmines from ground-penetrating radar signals. When a detection occurs at a given GPS coordinate, measures are taken at various depths in the soil. Each detection location is a bag containing feature vectors for different depths.

In (Maron & Lozano-Pérez, 1998), MIL is used to select stocks. Positive bags are created by pooling the 100 best-performing stocks each month, while negative bags contain the 5 worst performing stocks. An instance classifier selects the best stocks based on these bags.

In (Mcgovern & Jensen, 2003), a method learning relational structure in data predicts which movies will be nominated for an award. A movie is represented by a graph that models its relations to actors, studios, genre, release date, etc. The MIL algorithm identifies which subgraph explains the nomination to infer the success of test cases. This type of structural relation between bags and instance is akin to web page classification problems.

1.6 Experiments

In this section, 16 reference methods are compared using data sets that allows to shed in light on some of the problem characteristics discussed in Section 1.4. These experiments are conducted to show how problem characteristics influence the behavior of MIL algorithms, and demonstrate that these characteristics cannot be neglected when designing or comparing MIL algorithms. Four characteristics were selected, each from a different category, to represent the spectrum of characteristics. Algorithms are compared on the instance classification task, under different WR, with an unobservable negative distribution and with different degree of label noise. These characteristics were chosen because their effect can be isolated and easily parametrized. The reference methods used in the experiments were chosen because they represent a most families of approaches and include most of the most widely used reference methods. All experiments have been conducted using Matlab and some implementations from the MIL toolbox (Tax & Cheplygina, 2015) and the LAMDA website[1].

[1] http://lamda.nju.edu.cn/

Next we describe the reference methods used in the experiments. The methods are grouped based on the representation space following a taxonomy similar to (Amores, 2013). Instance-space methods classify each instance individually and combine the instance labels to assign a bag to a class. Bag-space methods do not classify, explicitly at least, instances individually. Bag-space methods employ one of two strategies: either compare distance between bags using an appropriate distance measure for sets or distributions, or encode the content of the bags to obtain a summarizing representation used in a supervised learning setting.

Instance-Space Methods

SI-SVM, SI-SVM-TH and SI-*k*NN: These are not MIL methods *per se*, but this type of approaches has been used as a reference point in several papers (Ray & Craven, 2005; Bunescu & Mooney, 2007b; Alpaydın *et al.*, 2015) to give an indication on the pertinence of using MIL methods instead of regular supervised algorithms. In these algorithms, each instance is assigned the label of its bag, and bag information is discarded. The classifier assigns a label to each instance, and a bag is positive if it contains at least one positive instance. For SI-SVM-TH the number of positive instances detected is compared to a threshold that is optimized on the training data.

MI-SVM and mi-SVM (Andrews *et al.*, 2002): These algorithms are transductive SVMs. Instances inherit their bag label. The SVM is trained and classifies each instance in the data set. It is then retrained using the new label assignments. This procedure is repeated until the labels remain stable. The resulting classifier is used to classify test instances. MI-SVM uses only the most positive instance of each bag for training, while mi-SVM uses all instances.

EM-DD (Zhang & Goldman, 2001): DD (Maron & Lozano-Pérez, 1998) measures the probability that a point in feature space belongs to the positive class given the class proportion of instances in the neighborhood. EM-DD uses Expectation-Maximization to locate the maximum of the DD function. Classification is based on the distance from this maximum point.

RSIS (Carbonneau *et al.*, 2016e): This method probabilistically identifies the witnesses in positive bags using a procedure based on random subspacing and clustering introduced in

(Carbonneau *et al.*, 2016c). Training subsets are sampled using the probabilistic labels of the instance to train an ensemble of SVM.

MIL-Boost (Babenko *et al.*, 2008): The MIL-Boost algorithm used in this paper is a generalization of the algorithm presented in (Viola *et al.*, 2006). The method is essentially the same as gradient boosting (Friedman, 2001) except that the loss function is based on bag classification error. The instances are classified individually, and their labels are combined to obtain bag labels.

Bag-Space Methods

C-kNN (Wang & Zucker, 2000): This is an adaptation of kNN to MIL problems. The distance between two bags is measured using the minimal Hausdorff distance. C-kNN relies on a two-level voting scheme inspired from the notion of citations and references in research papers. The algorithm was adapted in (Zhou *et al.*, 2005b) to perform instance classification.

MInD (Cheplygina *et al.*, 2015c): With this method, each bag is encoded by a vector whose fields are dissimilarities to the other bags in the training data set. A regular supervised classifier, an SVM in this case, classifies these feature vectors. Many dissimilarity measures are proposed in the paper, but the *meanmin* offered the best overall performance and will be used in this paper.

CCE (Zhou & Zhang, 2007): This algorithm is based on clustering and classifier ensembles. At first, the feature space is clustered using a fixed number of clusters. The bags are represented as binary vectors in which each bit corresponds to a cluster. A bit is set to 1 when at least one instance in a bag is assigned to its cluster. The binary codes are used to train one of the classifiers in the ensemble. Diversity is created in the ensemble by using a different number of clusters each time.

MILES (Chen *et al.*, 2006): In Multiple-Instance Learning via Embedded instance Selection (MILES) an SVM classifies bags represented by a feature vectors containing maximal similar-

ities to selected prototypes. The prototypes are instances from the training data selected by a 1-norm SVM. Instance classification relies on a score representing the instance contribution to the bag label.

NSK-SVM (Gärtner *et al.*, 2002): The normalized set kernel (NSK) basically averages the distances between all instances contained in two bags. The kernel is used in an SVM framework to perform bag classification.

miGraph (Zhou *et al.*, 2009): This method represents each bag by a graph in which instances correspond to nodes. Cliques are identified in the graph to adjust the instances weights. Instances belonging to large cliques have lower weight so that every concept present in the bag is equally represented when instances are averaged. A graph kernel captures similarity between bags and is used in an SVM.

BoW-SVM: Creating a dictionary of representative words is the first step when using a BoW method. This is achieved with BoW-SVM by performing k-means clustering on all the training instances (Amores, 2013). Next, instances are represented by the most similar word contained in the dictionary. Bags are represented by frequency histograms of the words. Histograms are classified by an SVM using a kernel suitable for histogram comparison (exponential χ^2 in this case).

EMD-SVM: The Earth Mover distance (EMD) (Rubner *et al.*, 2000) is a measure of the dissimilarity between two distributions. Each bag is a distribution of instances and the EMD is used to create a kernel used in an SVM.

1.6.1 Data Sets

Spatially Independent, Variable Area, and Lighting (SIVAL) (Rahmani *et al.*, 2005): This data set contains 500 images each segmented and manually labeled by (Settles *et al.*, 2008). It contains 25 classes of complex objects photographed from different viewpoints in various environments. Each bag is an image partitioned in approximately 30 segments. A 30-dimensional

feature vector encodes the color, texture and neighbor information of each segment. There are 60 images in each class, which are in turn considered as the positive class. 5 randomly selected images from each of the 24 other classes yield 120 negative bags. The WR is 25.5% in average but ranges from 3.1 to 90.6%. In this data set, unlike in other image data sets, co-occurrence information between the objects of interest and the background is nonexistent because all 25 objects are photographed in the same environment.

Birds (Briggs *et al.*, 2012): The bags of this data set correspond to 10 seconds recordings of bird songs from one or more species. The recording is segmented temporally to create instances, which belong to a particular bird or to background noises. These 10232 instances are represented by 38-dimensional feature vectors. Readers should refer to the original paper for details on the features. There are 13 types of bird in the data set, each in turn considered as the positive class. Therefore 13 problems are generated from this data set. In this data set, low WR poses a challenge, especially since it is not constant across bags. Moreover, bag classes are sometimes severely imbalanced.

Newsgroups (Settles *et al.*, 2008): The newsgroups data set was derived from the *20 Newsgroups* (Lang, 1995) data set corpus. It contains posts from newsgroups on 20 subjects. Each post is represented by 200-term frequency-inverse document frequency (TFIDF) features. This representation generally yields sparse vectors, in which each element is representative of a word frequency in the text scaled by its frequency in the entire corpus. When one of the subjects is selected as the positive class, all 19 other subjects are used as the negative class. The bags are collections of posts from different subjects. The positive bags contain an average of 3.7% of positive instances. This problem is semi-synthetic and does not correspond to a real-world application. There is thus no exploitable co-occurrence information, intra-bag similarities or bag structure. However, the representation yields sparse data, which is different from the two previous data sets, and is representative of text applications.

HEPMASS (Baldi *et al.*, 2016): The instances of this data set come from the HEPMASS Data Set[2]. It contains more than 10M instances which are simulation of particle collisions. The positive class correspond to collisions that produce exotic particles, while the negative class is background noise. Each instance is represented by a 27-dimensional feature vector containing low-level kinematic measurements and their combination to create higher level mass features (see original paper for more details). For each WR value, 10 versions of the MIL data are randomly generated. For each version, the training and a test sets contain 50 positive bags and 50 negative bags composed of 100 instances.

Letters (Frey & Slate, 1991): This semi-synthetic MIL data set uses instances from the Letter Recognition data set[3]. It contains a total of 20k instances representing each of the 26 letters in the English alphabet. Each of these letters can be seen as a concept and used to create different positive and negative distributions. Each letter is encoded by a 16-dimensional feature vector that has been standardized. The reader is referred to the original paper for more details. In WR experiments, for each WR value, 10 versions of the MIL data sets are randomly generated. Each version has a training and a test set. Both sets contain 50 positive bags and 50 negative bags each containing 20 instances. In the positive bags, witness are sampled from 3 letters randomly selected to represent positive concepts. All other letters are considered are negative concepts. For the experiments on negative class modeling, the data set is divided in train and test partitions each containing 200 bags. Each bag contains 20 instances. The bag classes are equally proportioned and the WR is 20%. Like before, the positive instances are samples from 3 randomly selected letters. Half of the remaining letters constitute the initial negative distribution and the other half constitutes the unknown negative distribution.

Gaussian Toy Data: In this synthetic data set, the positive instances are drawn from a 20-dimensional multivariate Gaussian distribution ($\mathcal{G}(\mu,\sigma)$) that represents the positive concept. The values of μ are drawn from $\mathcal{U}(-3,3)$. The covariance matrix (σ) is a randomly generated semi-definite positive matrix in which the diagonal values are scaled to $]0,0.1]$. The negative

[2] http://archive.ics.uci.edu/ml/datasets/HEPMASS

[3] https://archive.ics.uci.edu/ml/datasets/Letter+Recognition

instances are sampled from a randomly generated mixture of 10 similar Gaussian distributions. This distribution is gradually replaced by another randomly generated mixture. The data set is standardized after generation. The test and training partitions both contain 100 bags. There are 20 instances in each bag and the WR is 20%.

1.6.2 Instance-Level Classification

In this section, the reference methods with instance classification capabilities will be compared on three benchmark data sets: SIVAL, Birds and Newsgroups. These data sets are selected because they represent three different application fields and because instance labels are provided, which is somewhat uncommon with MIL benchmark data sets. There already exist several comparative studies for bag-level classification, we refer interested reader to (Amores, 2013; Kandemir & Hamprecht, 2015).

The experiments were conducted using a nested cross-fold validation protocol (Stone, 1974). It consists of two cross-validation loops. An outer loop assesses the performance of the algorithm in test, and an inner loop is used to optimize the algorithm hyper-parameters. This means that for each test fold of the outer loop, hyper-parameters optimization is performed via grid-search. Average performance is reported on results for the outer loop test folds.

Instance classification problems often exhibit class imbalance, especially when the WR is small. In these cases, comparing algorithm in terms of accuracy can be misleading. In this section, algorithms are compared in terms of unweighted average recall (UAR) and F_1-score. The UAR is the average of the accuracy for each class. The F_1-score is the harmonic mean between precision and recall. The 3 data sets translate into 58 different problems. For easy comparison, Fig. 1.7 and 1.8 present the results in the form of critical difference diagrams (Demsar, 2006) with a significance level of 1%.

Results indicate that a successful strategy for instance classification is to discard bag information. With both metrics, the best algorithms are mi-SVM and SI-SVM, which assign the bag label to each instance and then treat them as atomic elements. This is consistent to

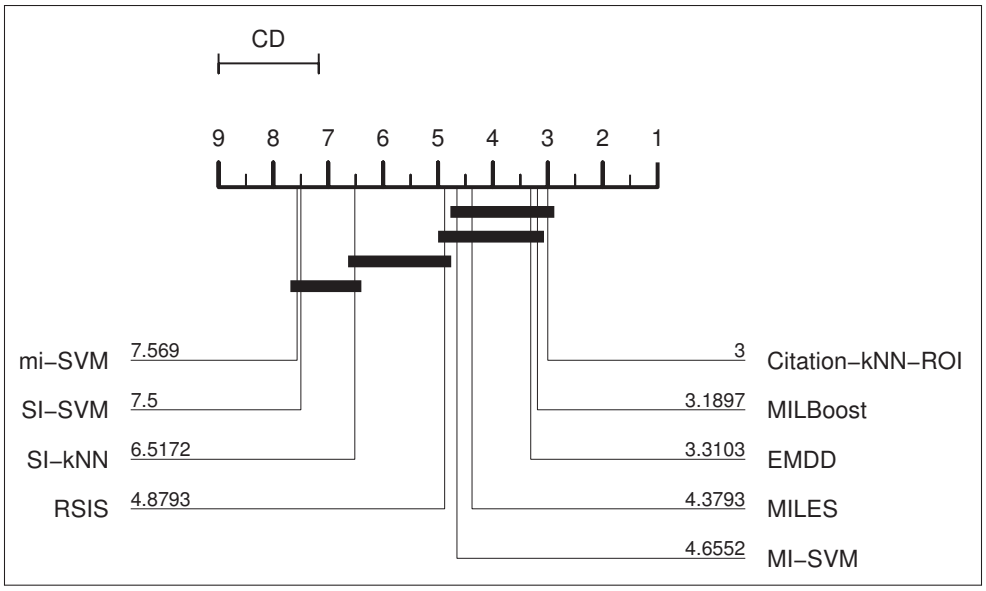


Figure 1.7 Critical difference diagram for UAR on instance classification ($\alpha = 0.01$). Higher numbers are better

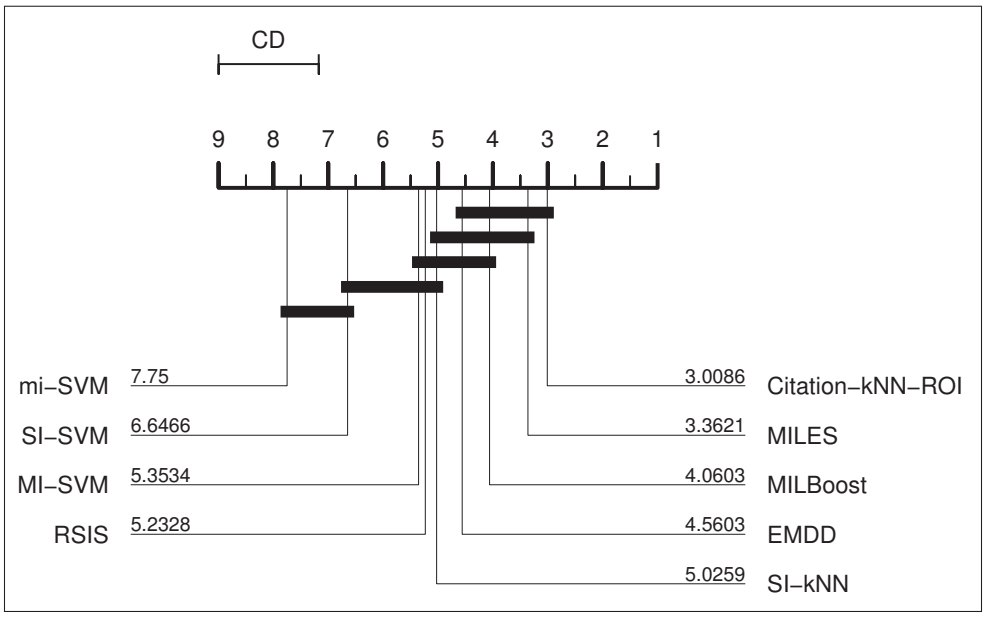


Figure 1.8 Critical difference diagram for the F_1-score on instance classification ($\alpha = 0.01$). Higher numbers are better

the results obtained in (Kandemir & Hamprecht, 2015). These two methods are closely related because SI-SVM corresponds to the first iteration of mi-SVM. SI-kNN also yield competitive

results and uses the same strategy. Even if the Birds and the Newsgroups data sets both possess low WR, it would seem that supervised methods are better suited for this task than MIL methods which use bag accuracy as an optimization objective (MILES, EMDD and MIL Boost). MI-SVM and RSIS rely on the identification of the most positive instances in each bag. This strategy seems successful to some degree, but is prone to ignore more ambiguous positive instances that are dominated by the others in the same bag. These conclusions have also been observed in the results obtained on the individual data sets.

1.6.3 Bag Composition: Witness Rate

These experiments study the effects of the WR on MIL algorithm performances. Two semi-synthetic data sets were created to allow control over the WR, and observe the behavior of the reference methods in greater detail: Letters and HEPMASS. These data sets are created from supervised problems that were artificially arranged in bags. This has the advantage of eliminating any structure and co-occurrence in the data, and thus better isolate the effect of WR. The original data sets must possess a high number of instances to emulate low WR. In the Letters data set, the positive class contains three concepts while in HEPMASS there is only one concept, which has an impact for some algorithms.

All hyper-parameters were optimized for each version of the data sets, and for each WR value using grid search and cross-validation. The results reported in Fig. 1.9, 1.10, 1.11 and 1.12 are the average results obtained on the test data for each of the 10 generated versions. Performance are compared using AUC and the UAR.

There are several things that can be concluded by examining the experiment results. Firstly, **for all methods, lower WR translates into lower accuracy**. However, Fig. 1.9 shows that **for the instance classification task, higher WR does not necessarily means higher accuracy** for all methods. In fact, for the Letters data set, three different letters are used to create positive instances which makes the positive distribution multimodal. As discussed in Section 1.6.2, some methods are optimized for bag classification (EM-DD, MI-SVM, MILES, MILBoost, RSIS-

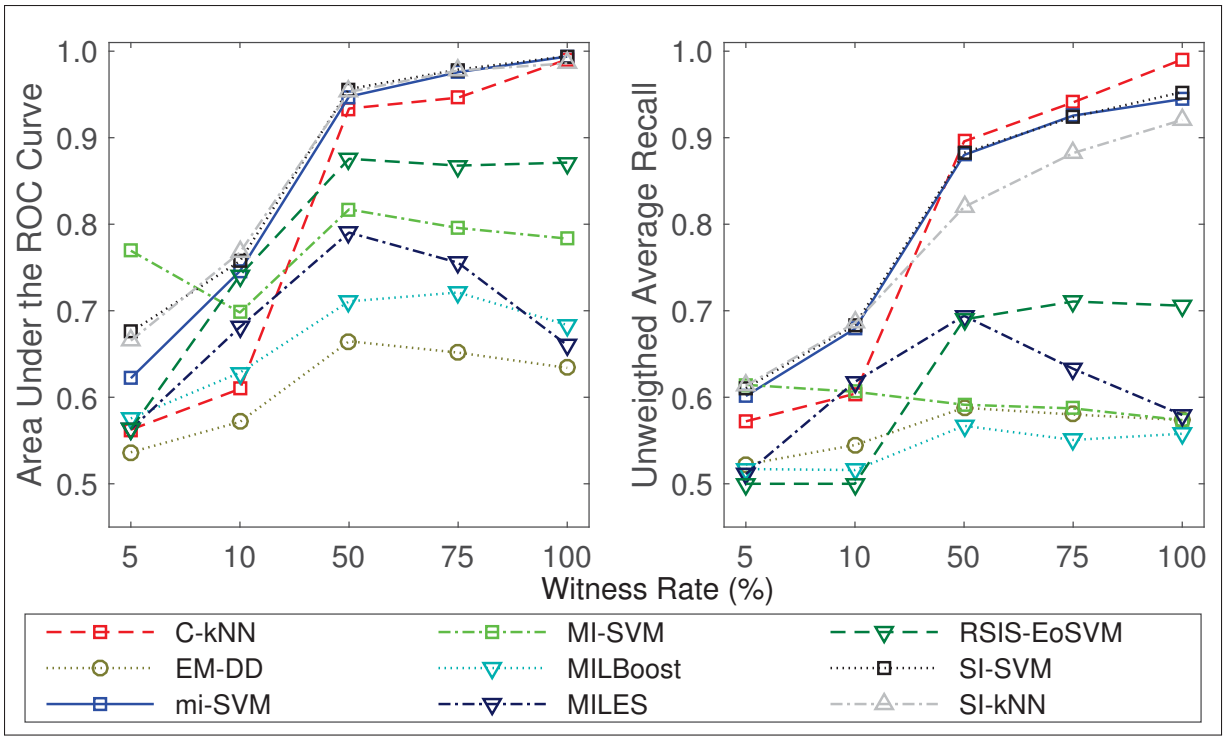


Figure 1.9 Average performance of the MIL algorithms for instance classification on the Letters data set as the witness rate increases

EoSVM). In those cases, once a letter is assigned to the positive class in a positive bag, the bag is correctly classified. The remaining positive letters can be ignored and the algorithm still achieves perfect bag classification. This can be observed by comparing Fig. 1.9 and 1.11 with Fig. 1.10 and 1.12, where the methods optimized for bag classification deliver lower accuracy for instance classification, but their accuracy is comparable to other instance-based methods when classifying bags. This explains in part the observation (Doran & Ray, 2014a; Vanwinckelen *et al.*, 2015) that an algorithm performance for one task is not always representative of the performance in the other.

The results in Fig. 1.9 and 1.11 suggest that **supervised classifiers are as effective for instance classification as the best MIL classifiers when the WR is over 50%**. In this case, the mislabeled negative instance are just noise in the training set, which is easily dealt with by the SVM or the voting scheme of the SI-kNN. Even when WR is lower than 50% supervised methods perform better than some of their MIL counterparts. MI-SVM has higher AUC per-

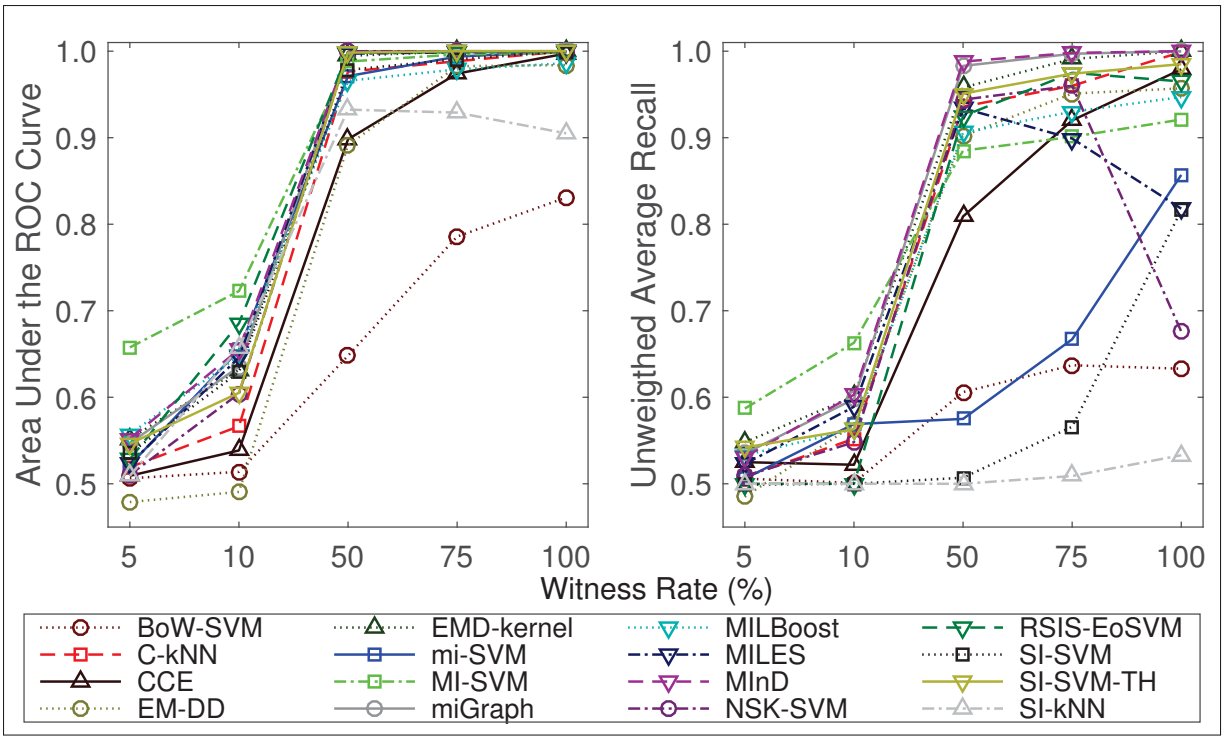


Figure 1.10 Average performance of the MIL algorithms for bag classification on the Letters data set as the witness rate increases

formance when the WR is at its lowest compared to the other method. This is explained by the fact that positive bags are represented by their single most positive instance. When the WR is at its minimum, there is only one witness per bag which coincides with this representation.

Table 1.2 Ranking of instance-based methods *vs*. bag-based methods for the bag classification task

		WR	
Metric	**Method Type**	$< 50\%$	$\geq 50\%$
Mean Rank (AUC)	Instance-based	9.3	11.3
	Bag-based	7.7	5.7
Mean rank (UAR)	Instance-based	10.0	11.0
	Bag-based	7.0	6.0

The results for bag classification are reported in Fig. 1.10 and 1.12. For an easier comparison between instance- and bag-based methods, mean ranks for all experiments are reported in Table

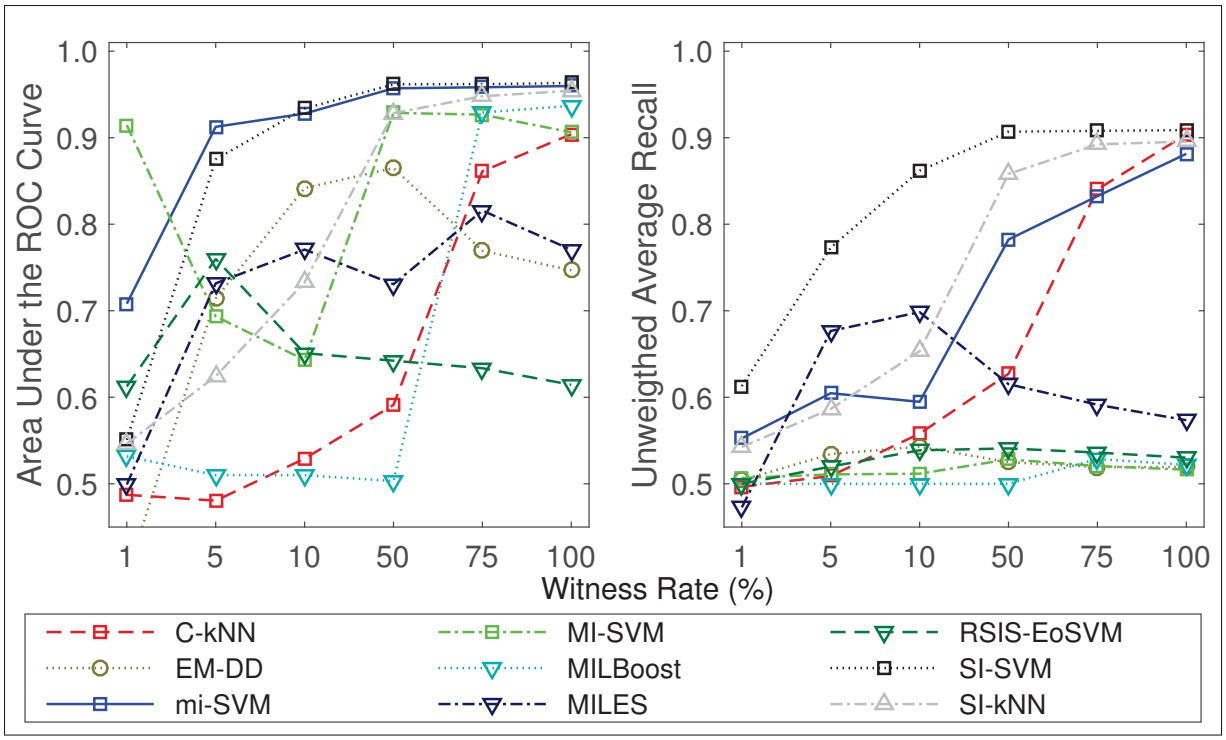


Figure 1.11 Average performance of the MIL algorithms for instance classification on the HEPMASS data set as the witness rate increases

1.2. These results show that, **in general, bag-space methods outperform their instance-space counterparts at higher WR** ($\geq 50\%$). At lower WR ($5 \sim 10\%$), the difference between both approaches is lower. However, in the Letters experiment, MI-SVM outperform all other methods by a significant margin, while in the HEPMASS experiment, EMD-SVM and NSK-SVM perform better. This suggests that **at lower WRs, there are other factors to consider when selecting a method**, such as the shape of the positive and negative distributions and the consistency of the WR across positive bags.

1.6.4 Data Distribution: Non-Representative Negative Distribution

In some applications, the negative instance distribution cannot be entirely represented by the training data set. The experiments in this section measure the ability of MIL algorithms to deal with a negative distribution different in test and training. We use two data sets in these experiments: the Letters data set and the synthetic Gaussian toy data set created specially for

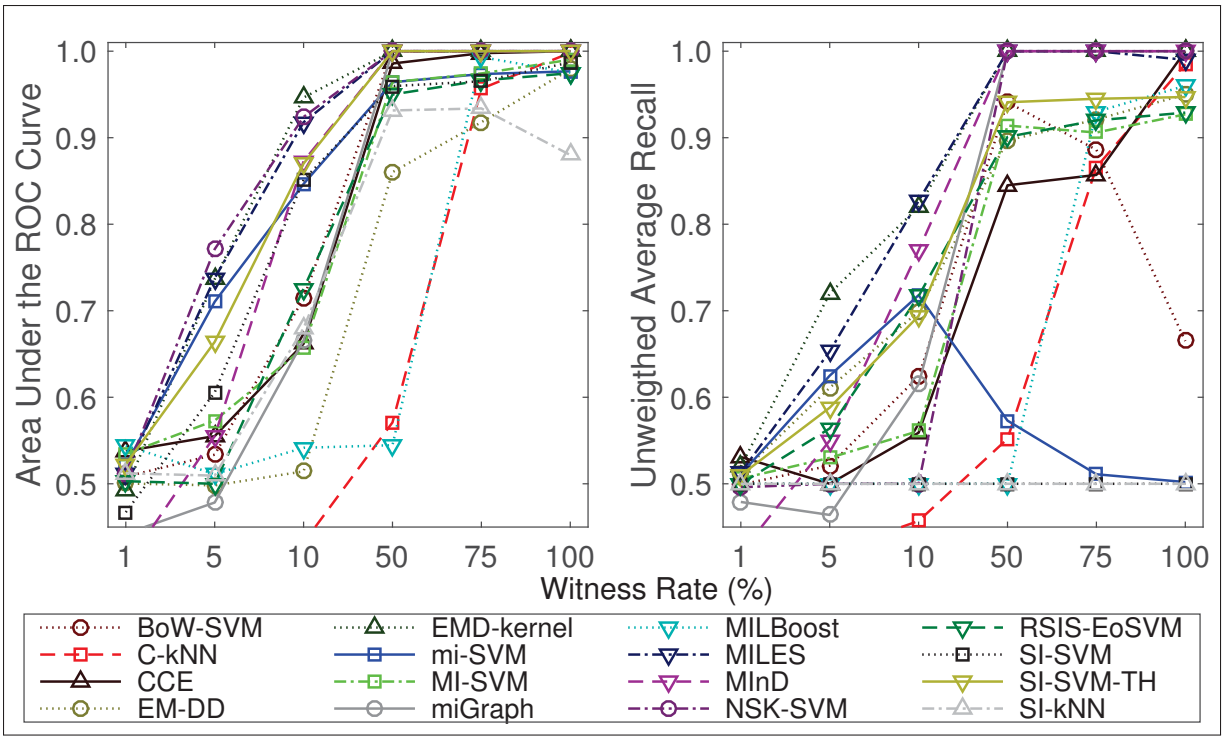


Figure 1.12 Average performance of the MIL algorithms for bag classification on the HEPMASS data set as the witness rate increases

this experiment. Using these two data sets makes it possible to control factors to measure the effect of a changing negative distribution in isolation from other problem characteristics. In each experiment, there are two different negative instance distributions. The first one is used to generate the training data. For the test data sets, at first, the negative instances are also sampled from this same distribution, but are gradually replaced by instances from the second distribution. The positive instances are sampled from the same distribution in both the training and test sets. For instance, using the Letters data set, this means that in the training data set the letter A, B and C are used as negative instances. Gradually, the instance from A, B and C are replaced by instance on the letter D, E and F.

The results of the experiments, illustrated in Fig. 1.13, 1.14, 1.15 and 1.16, show that **most algorithms have decreasing performance when the test negative instances distribution differs from the training distribution**. However, C-kNN exhibits a contrasting behavior. More the test instances differ from test to training, the better are performances. This is because

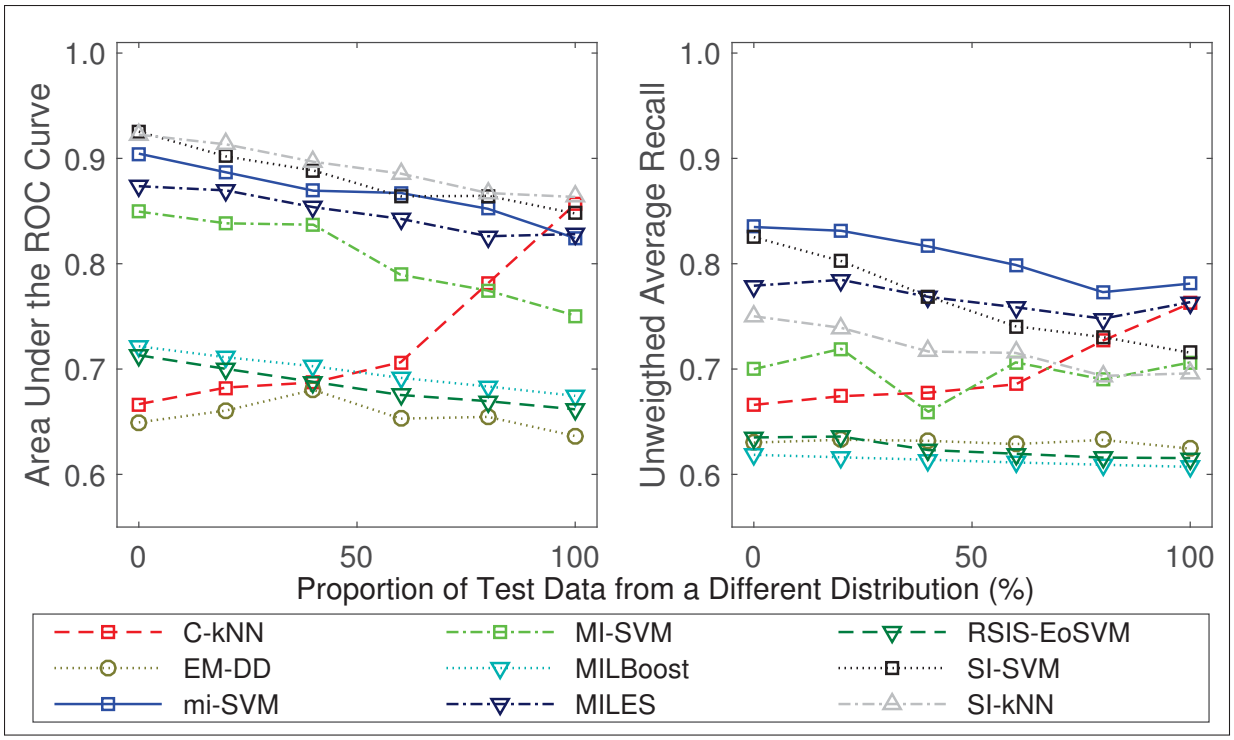


Figure 1.13 Average performance for instance classification on the Letters data as the test negative instance distribution increasingly differs from the training distribution

C-kNN uses the minimal Hausdorff distance as a similarity metric between bags. This is the distance between the two closest instances from each bag. If the negative instances come from the same distribution in all the bags, it is likely that the closest instance are both from the negative distribution, even if the bags are positive. If the bags have different labels, this leads to misclassification. If the negative test instances are different from those in the training set, the distance between two negative instances is likely to be greater than the distance between two positive instances, which are from the same distribution in both sets. Thus, positive bags are found to be closer to other positive bags leading to a higher accuracy.

The results for both data sets suggest that **bag-space methods are better for dealing with new negative distributions**. This may contribute to their success in computer vision applications. In Fig. 1.14 the AUC for bag classification is stable for most method while their accuracy decreases. This suggest that the score functions learned by the algorithms are still suitable for

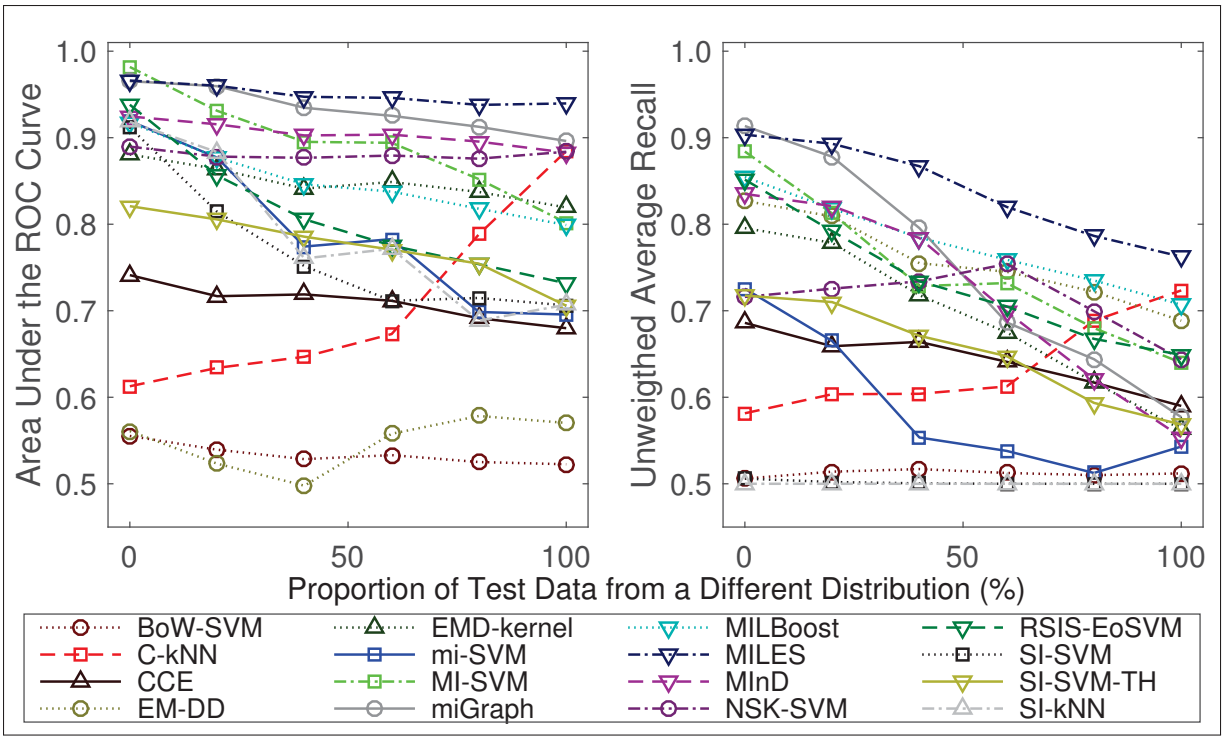


Figure 1.14 Average performance for bag classification on the Letters data as the test negative instance distribution increasingly differs from the training distribution

the new distribution, but the thresholds should be adjusted. This observation motivates the use of adaptive methods in practice which would adjust the decision threshold as new data arrives.

1.6.5 Label Ambiguity: Label Noise

It is generally assumed that the weak supervision provided by bag labels is accurate. However, as explained in Section 1.4.4, this is not always the case. Here, we measure the ability of reference algorithms to deal with noisy labels. Experiments are conducted on the Letters and SIVAL datasets. In these experiments, an increasing proportion of bag labels in the training set are inverted. When 50% of the labels are inverted, both classes contain an equal proportion of true positive and negative bags. After, 50% of the labels are inverted, the problem can be seen as the same classification problem where the negative class is considered as the positive class.

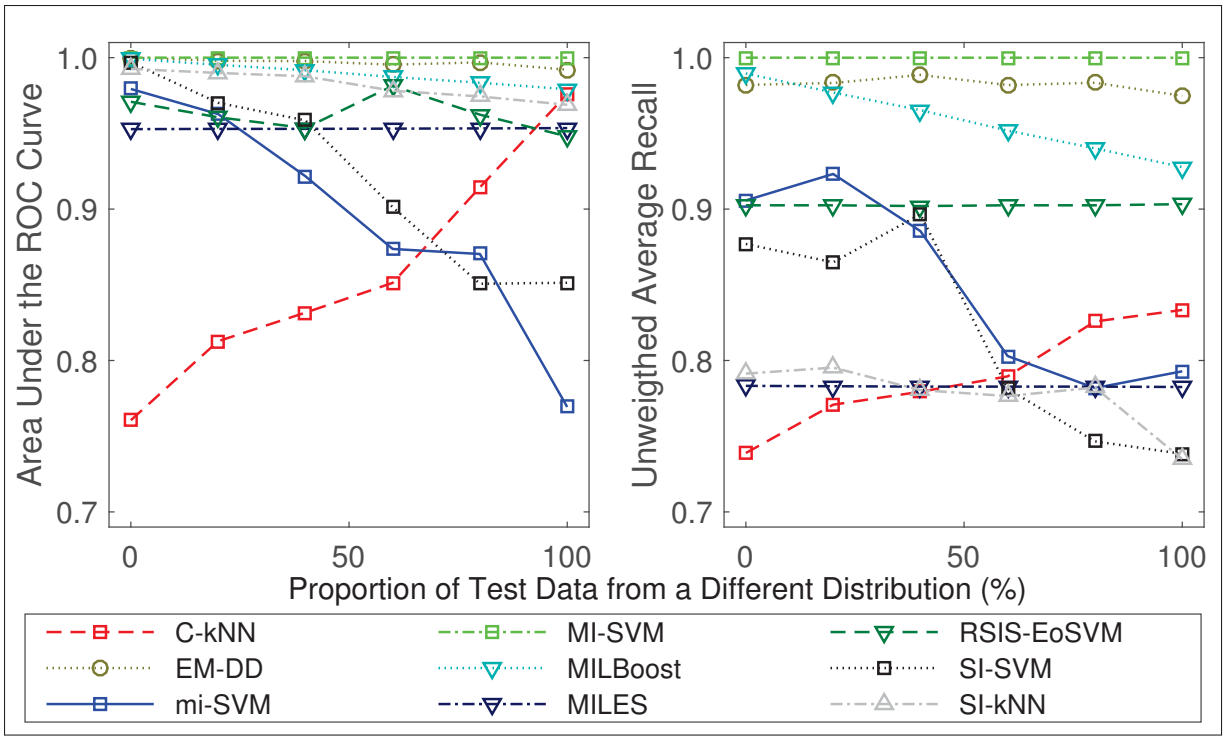


Figure 1.15 Average performance for instance classification on Gaussian toy data as the test negative instance distribution increasingly differs from the training distribution

For bag classification, the experiments reveal that label noise robustness relates to the decision space used by MIL classifiers. **Bag-space methods using an embedding strategy (e.g. EMD-kernel, miGraph, MInD) are the most robust to label noise.** The results for these methods are reported in Fig. 1.19 and 1.20. The symmetry in their performance curves suggests that these **embedding methods make no distinction between the positive and the negative class**, and thus their label can be interchanged seamlessly. Embedding algorithms encode bags in a single feature vector and view the bag classification problem as a supervised problem. In that regard, the robustness of the method depends on the type of classifier used by a given method.

All methods in this experiment use an SVM which is known to be vulnerable to label noise (Frenay & Verleysen, 2014). Since all classifiers are SVMs, it is easier to compare embedding techniques. The performance curve shapes show which type of embedding is the most noise resistant. MInD and EMD-kernel both maintain their level of performance until there is 30% of mislabeled bags, while the performance of MILES, NSK-SVM and miGraph steadily decrease

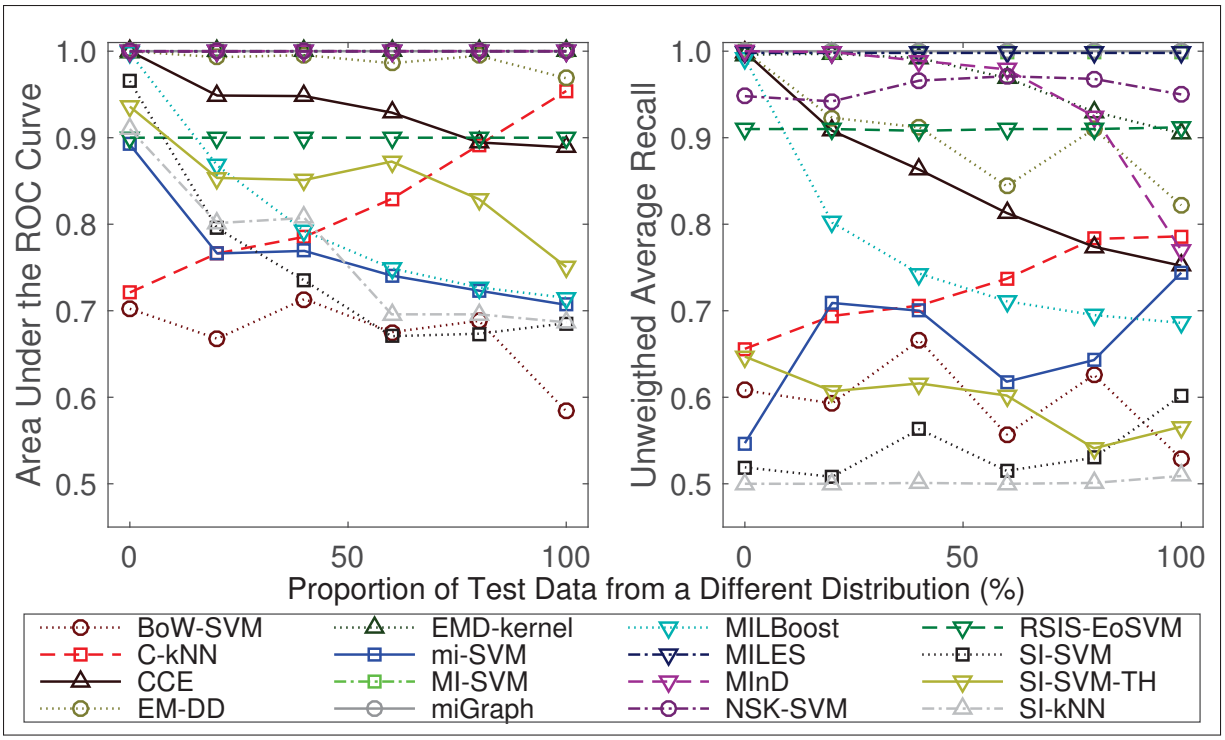


Figure 1.16 Average performance for bag classification on Gaussian toy data as the test negative instance distribution increasingly differs from the training distribution

as the noise increases. MInD and EMD-kernel describe bags as distances between the other bags in a kernel. EMD-kernel computes the distance between distribution of instances, while MInD averages the minimal distance between all instances, which can also be seen as a distance between the two distributions. CCE also represent instance distribution in a bag and exhibited a similar noise resistance is the experiments on SIVAL. Based on these observations, it would seem that **characterizing bags as instance distributions is a successful strategy to deal with label noise.**

While embedding methods characterize the distribution of instances in bags, MIL methods working under the standard MIL assumption (e.g. mi-SVM, MILBoost and MI-SVM) use a different approach. These **instance-space methods learn to identify witnesses as a step toward bag classification. In that case, the positive and the negative class are not equivalent.** This is shown by the asymmetry of the performance curves in Fig. 1.21 and 1.22. For most of these methods, when a majority of labels are inverted performance tends towards random clas-

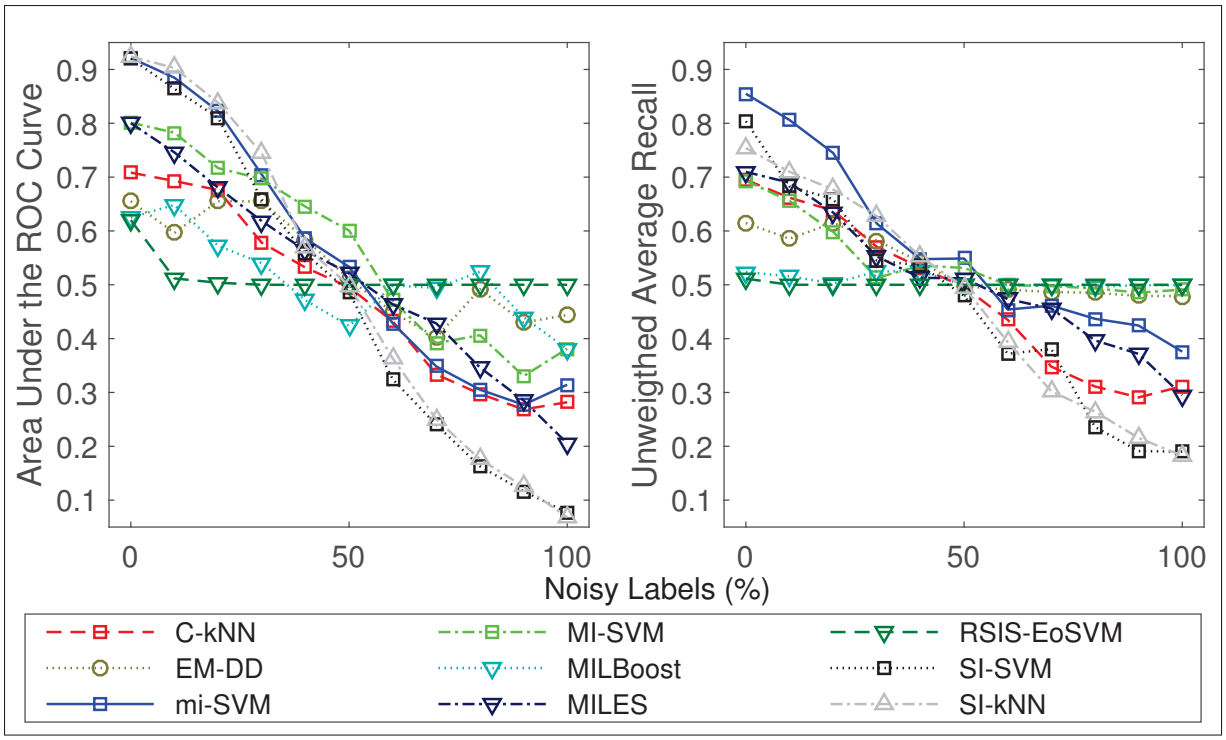


Figure 1.17 Average performance of the MIL algorithms for instance classification on the Letters data with increasing label noise

sification. For instance-space methods, positive concepts must be cohesive and shared between positive bags while excluded from negative bags. When positive bags are mislabeled, positive instances are found in negative bags which makes the identification of the positive concept difficult. This is why **instance-space methods are more vulnerable to noise**. As shown in Fig. 1.21 and 1.22, the performance of all methods steadily degrades if the label noise level is over 10%. This is related to the instance classification performance degradation observed in Fig. 1.17 and 1.18. The experiments did not reveal a strategy that is more noise resistant than the others for instance classification.

In a nutshell, bag-space and instance-space methods differ in their dependency on the identification of positive concept. This identification process highly relies on the correctness of the bag labels which hinders the performance of instance-space method in noisy problems.

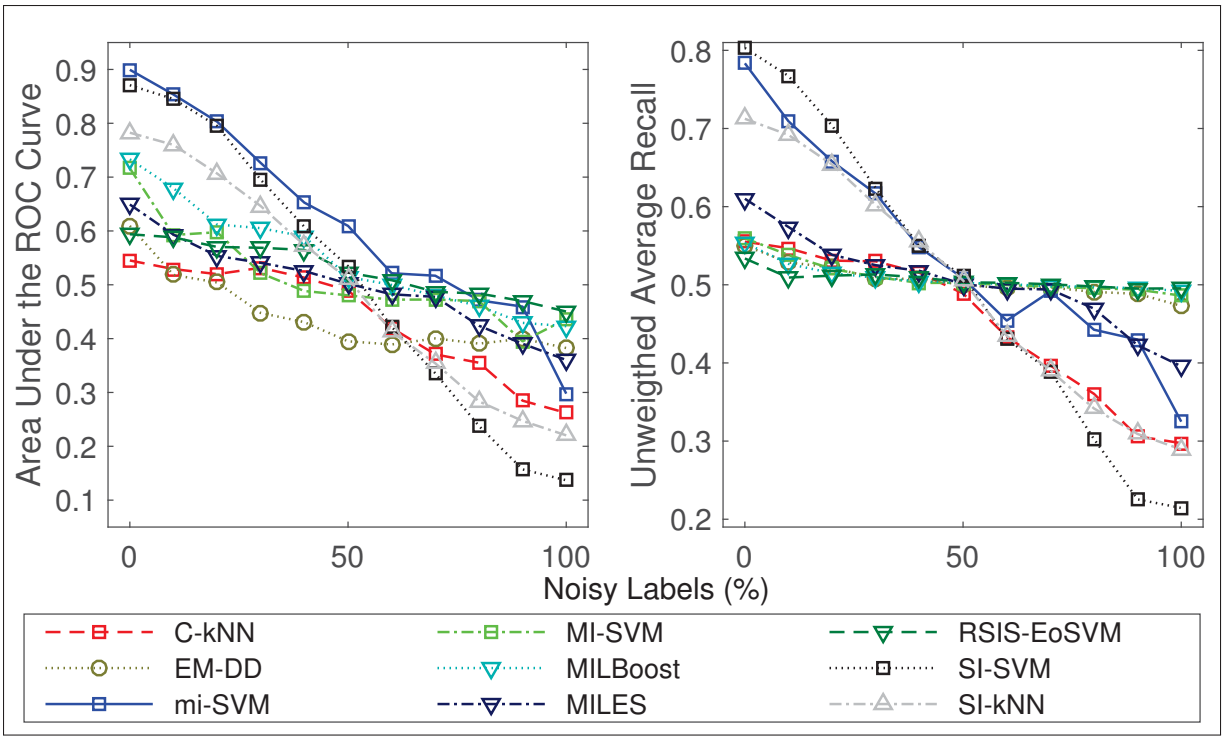


Figure 1.18 Average performance of the MIL algorithms for instance classification on the SIVAL data with increasing label noise

1.7 Discussion

The problem characteristics identified in this paper allow for a discussion on validation procedures of MIL algorithms. These suggestions are also based on the observations from the experiments in the previous section. Next we discuss practical considerations for MIL like available softwares and the complexity of MIL methods. Then, we identify interesting research avenues for MIL.

1.7.1 Benchmarks Data Sets

Several characteristics inherent to MIL problems were discussed in this paper. Experiments confirmed what has been observed by many researchers before: algorithms perform differently depending on the type of MIL problem, and several characteristics define a MIL problem. However, even to this day, many approaches are validated only with the Musk and Tiger/Ele-

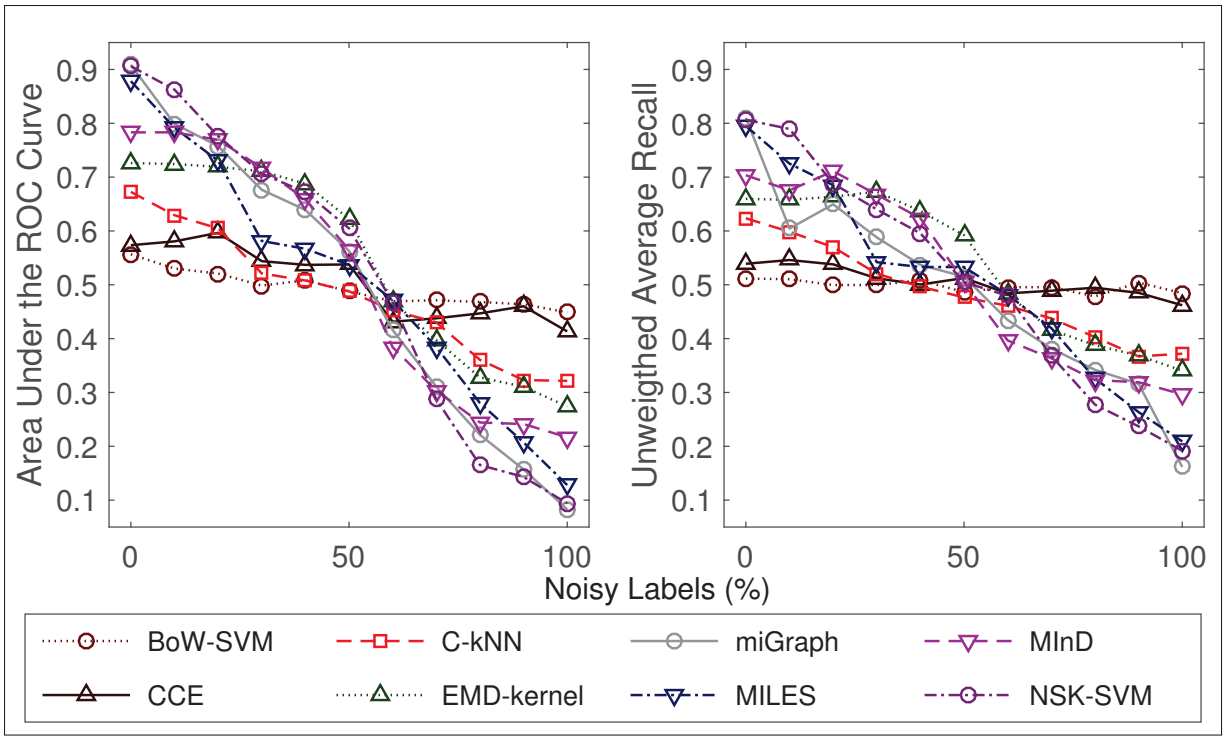


Figure 1.19 Average performance of the bag-space MIL algorithms for bag classification on the Letters data with increasing label noise

phant/Fox (TEF) data sets. There are several problems with these benchmark data sets. First, they pose only some of the challenges discussed earlier. For example, the WR of these data sets is high. Since the instance labels are not supplied, the real WR is unknown. However, it has been estimated in some papers (Li & Sminchisescu, 2010; Li *et al.*, 2013; Gehler & Chapelle, 2007) which reported 82 to 100% for Musk1, 23 to 90% for Musk2 and 38 to 100% for TEF. Moreover, in the Musk data sets, there is no explicit structure to be exploited. In the TEF data sets, the instances are represented by 230-dimensional feature vectors characterizing by color, texture and shape descriptors. No further details are given on these features, except that this representation is sub-optimal and should be further investigated (Andrews *et al.*, 2002). It is possible that the theoretical Bayesian error has already been reached for this feature representation and that better results are obtained on account of protocol related technicality, such as fold partitions. Also, since the annotations at instance level are not available, it is difficult to assess if the fox classifier really identifies foxes, or if it identifies background elements related

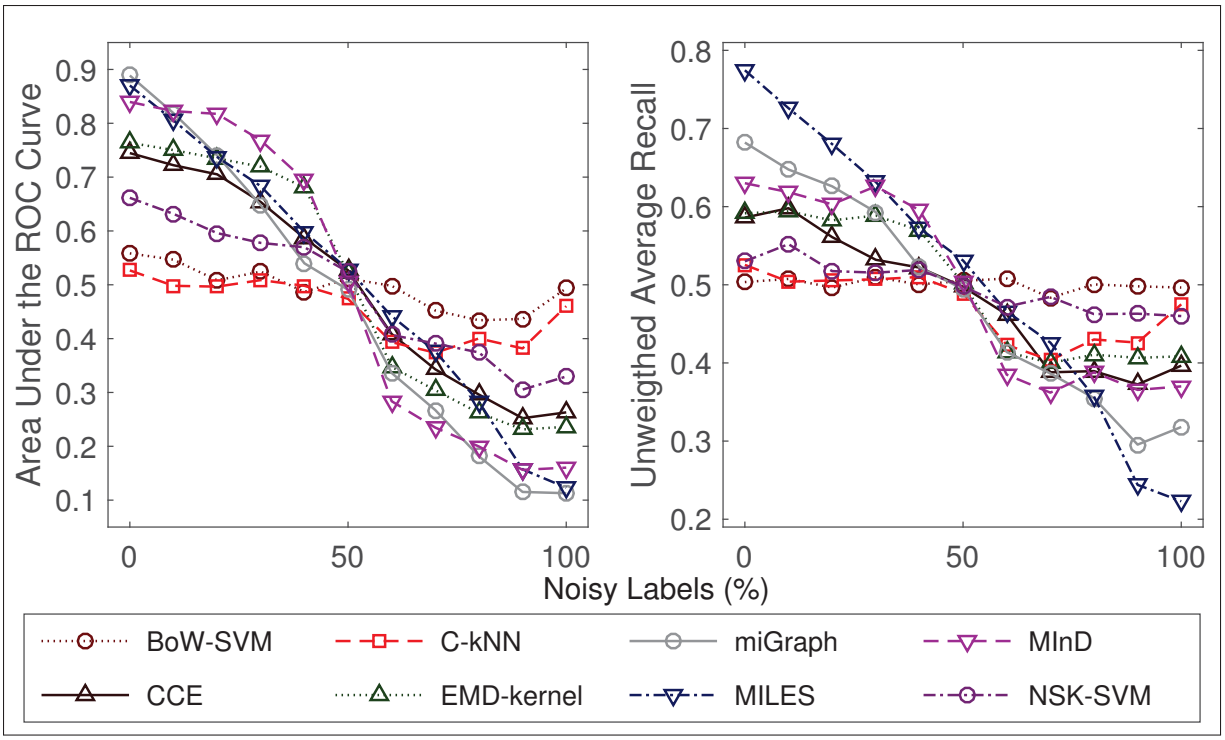


Figure 1.20 Average performance of the bag-space MIL algorithms for bag classification on the SIVAL data with increasing label noise

to foxes such as forest segments. This would explain the high WR estimated in (Li & Sminchisescu, 2010; Li *et al.*, 2013; Gehler & Chapelle, 2007). Since the state-of-the-art accuracy on this class is around 70%, it is plausible that a large proportion of the animals in the negative class live in deserts or under the sea. For all these reasons, in our opinion, while the Musk and TEF data sets are representative of some problems, using more diverse benchmarks would provide a more meaningful comparison of MIL algorithms.

Because of the aforementioned TEF shortcomings, researchers should use more appropriate benchmark data for computer vision tasks. For example, several methods have been compared on the SIVAL data set. It contains different objects captured in the same environments, and provides labels for instances. In each image the objects of interest are segmented into several parts. The algorithms ability to leverage co-occurrence can thus be measured, and since the objects are all captured in the same environments, the background instances do not interfere in the classification process. However, it would be more beneficial for the MIL community to

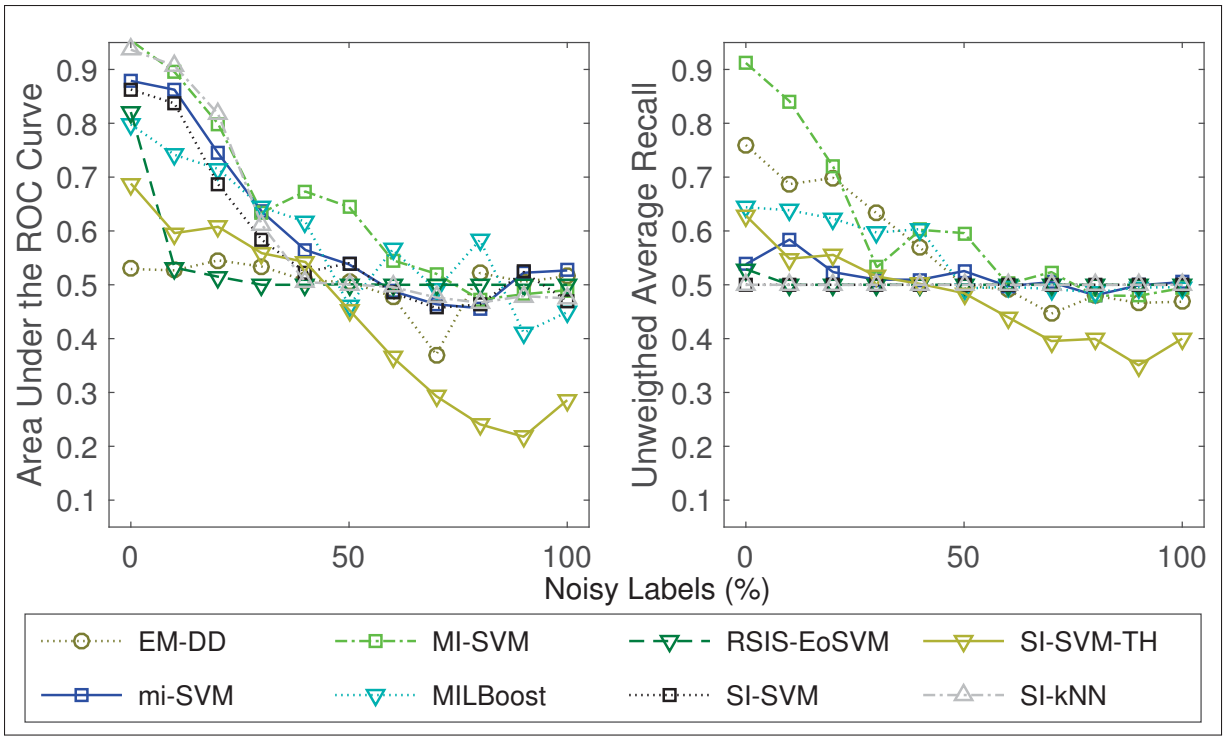


Figure 1.21 Average performance of the instance-space MIL algorithms for bag classification on the Letters data with increasing label noise

use other existing strongly annotated computer vision data sets (e.g. Pascal VOC (Everingham *et al.*, 2010) or ImageNet (Russakovsky *et al.*, 2015)) as benchmarks. These types of data set provide bounding box or even pixel-level annotations that can be used to create instance labels in MIL problems. MIL algorithms could be compared to other types of techniques, which is almost never done in the MIL literature. Also, supplying the position of instances in images for these new computer vision MIL benchmarks would help to develop and compare methods that leverage spatial structure in bags.

In application fields other than computer vision, there are relatively few publicly available real-world data sets. From these few data sets, to our knowledge, there is only one (Birds (Briggs *et al.*, 2012)) that supply instance labels and is non-artificial. This is understandable since MIL is often used to avoid the labor-intensive instance labeling process. Nevertheless, real-world MIL data needs to be created to measure the instance labeling capability of different MIL methods, as it is an increasingly important task. Also, to our knowledge, there is no publicly

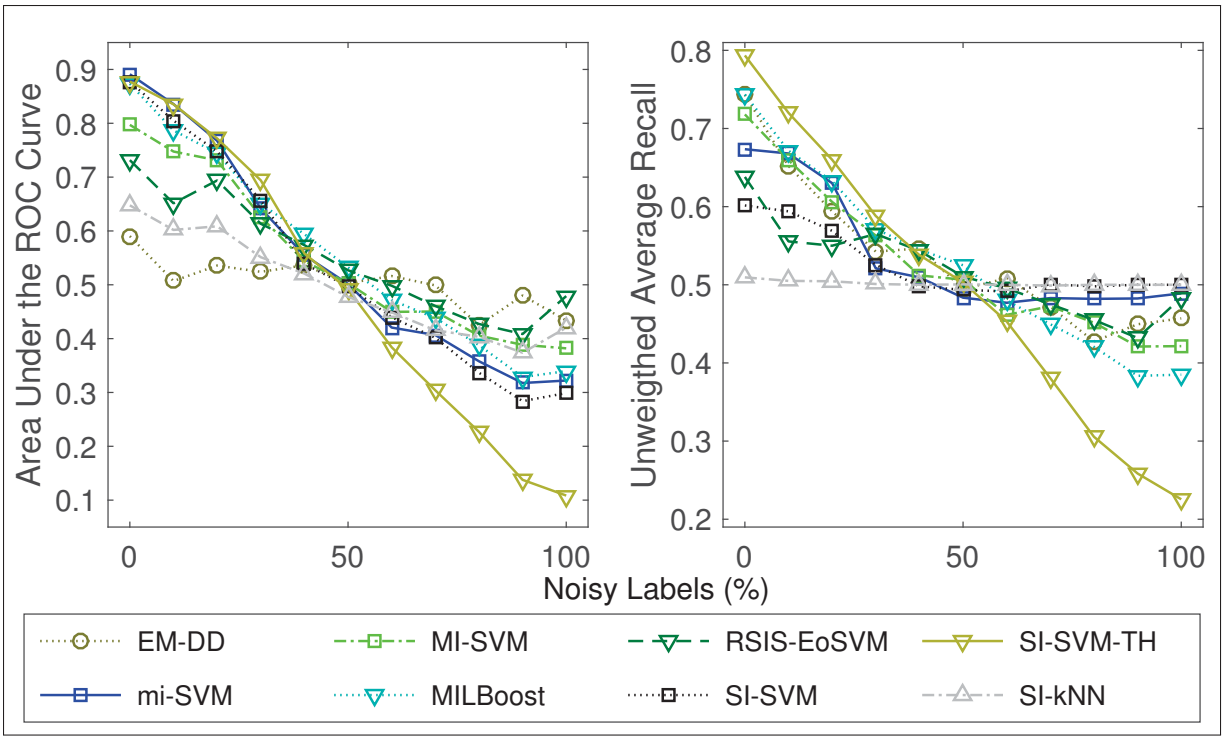


Figure 1.22 Average performance of the instance-space MIL algorithms for bag classification on the SIVAL data with increasing label noise

available benchmark data set for MIL regression, which would surely stimulate research on this task.

Finally, several methods are validated using semi-artificial data sets. These data sets are useful to isolate one parameter of MIL problems, but are generally not representative of real-world data. In these data sets, instances are usually i.i.d. which almost never happens in real problems. Authors should justify the use of this type of data, clearly mention what assumptions are made and how the data sets are different from real data. As a start, Table 1.3 compiles the characteristics which are believed to be associated with some of the most widely used benchmark data sets, based on parameter estimation and data descriptions found in literature. These are believed to be true but would benefit from rigorous investigation in the future.

In short, whenever only the Musk and the TEF data sets are used to validate a new method, it is difficult to predict how the methods will perform in different MIL problems. Moreover,

because researchers are encouraged to evaluate their methods on these data sets, promising models may be dismissed too early because they do not outperform the best performing methods optimized on these benchmark data sets. We argue that a better understanding of the characteristics of the MIL data sets should be promoted, and that the community should use other data sets to compare MIL algorithms in regard of the challenges and properties of MIL problems.

Table 1.3 Table compiling the characteristics of MIL benchmark data sets based on statement in the literature

Benchmark MIL Data Sets	**Instance labels**	**Low witness rate**	**Intra-bag similarities**	**Instance co-occurence**	**Structure in bags**	**Multimodal positive distribution**	**Non-representative negative distribution**	**Label noise**	**Semi-Artificial**
Musk (Dietterich *et al.*, 1997)			✓			✓	✓		
Tiger, Fox, Elephant (Andrews *et al.*, 2002)			✓	✓		✓	✓		
SIVAL (Settles *et al.*, 2008)	✓					✓			
Birds (Briggs *et al.*, 2012)	✓	✓		✓					
Newsgroups (Settles *et al.*, 2008)	✓	✓				✓			✓
Corel (Chen & Wang, 2004)			✓	✓		✓	✓		
Messidor Diabetic Retinopathy (Kandemir & Hamprecht, 2015)		✓				✓			
UCSB Breast (Kandemir *et al.*, 2014b)		✓				✓			
Biocreative (Ray & Craven, 2005)			✓	✓		✓			

1.7.2 Accuracy vs. AUC

While benchmark data is of paramount importance, the proper selection of performance metrics is equally important to avoid hasty conclusions. In all experiments, some algorithms have obtained contrasting performance when comparing AUC to accuracy and UAR. This has also been observed in other experiments (Carbonneau *et al.*, 2016e). This is an important factor that must be taken into consideration when comparing MIL algorithms.

Some algorithms (e.g. mi-SVM, SI-kNN, SI-SVM, miGraph, MILES) obtain high AUC that does not translate into high accuracy. There may be many reasons for this. Some algorithms optimize the decision thresholds based on bag accuracy, while others infer individual instance labels. In the first case, the algorithm is more prone to FN, while the latter is more prone to FP because of the asymmetric misclassification costs discussed in Section 1.6.2. Figure 1.14 and Figure 1.16 in Section 1.6.4 clearly illustrate this. As the negative distribution changes, the AUC remains stable for many algorithm, while accuracy decreases (e.g. miGraph, MILES, BoW-SVM). This means that the score function was still suitable for classification, but the decision threshold was no longer optimal. Considering the right end of the AUC curves in Figure 1.14, where negative instances are completely sampled from a new distribution, one could conclude that miGraph performs better than RSIS-EoSVM. However, when comparing with UAR, the inverse can be concluded. One could argue the AUC is a sufficient performance metric assuming that the decision threshold is optimized on a validation set, however, in many problems, the amount of available data is too limited for this assumption to hold. Also in the case of instance classification, instance labels are unknown, therefore, it is not possible to perform such optimization.

In our opinion, the algorithms ability to accurately set this threshold is an important characteristic that should be measured, as well as the ability to learn a suitable score function. Therefore, accuracy measures (e.g. accuracy, F_1-score, etc.) should always be reported alongside AUC.

1.7.3 Open Source Toolboxes

We think it is a good practice to report results from original papers because each method has been optimized by its own author for maximal performance. Some authors have published their code to allow fellow researchers to conduct more extensive experiments with their methods on other data sets. There are already several methods available from author websites (Vanwinckelen *et al.*, 2015; Carbonneau *et al.*, 2016e; Kandemir & Hamprecht, 2015; Gehler & Chapelle, 2007; Chen & Wang, 2004; Settles *et al.*, 2008). The website of the LAMDA[4] lab is worth

[4] bhttp://lamda.nju.edu.cn

mentioning as it contains several implementations of MIL methods for Matlab. Other Matlab implementations of reference MIL methods can be found in the MIL toolbox (Tax & Cheplygina, 2015). There are also machine learning and data mining software packages such as Weka (Frank *et al.*, 2016), KEEL (Alcala-Fdez *et al.*, 2011) and JCLEC (Ventura *et al.*, 2008) for which MIL modules exist. Finally, the Python implementations of SVM-based MIL algorithms used in (Doran & Ray, 2014a) are also available on-line. The wide variety of MIL problems calls for more comparative studies which will be facilitated by the availability of readily usable code. In that spirit, the code we used in our experiments have been made available on-line[5].

1.7.4 Computational Complexity

It has been noted by several authors that many MIL algorithms are too computationally expensive to be used with large data sets (Fung *et al.*, 2007; Amores, 2013). This represents a serious problem since one of the advantages of MIL is to increase the quantity of data available for training by leveraging weakly labeled data.

Many algorithms in literature do not scale well to big data sets. For example, the computational complexity of an SVM is between $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$ when using traditional QP and LP solvers (Bottou *et al.*, 2007), where n is the number of instances. Thus, many methods using SVM and SVM-like algorithms (Chen *et al.*, 2006; Andrews *et al.*, 2002; Bunescu & Mooney, 2007b; Fung *et al.*, 2007; Bergeron *et al.*, 2008; Mangasarian & Wild, 2008) rapidly become impractical as the number of instances increases (Bergeron *et al.*, 2012). To address this problem, in (Bergeron *et al.*, 2012), a bundle algorithm (Fuduli *et al.*, 2003) is used to solve the SVM optimization problem in linear time ($\mathcal{O}(n)$). Alternatively, it has been proposed to use gradient descent with logistic regressions in a MILES like algorithm Fu & Robles-Kelly (2008). Gradient descent algorithms is more appropriate for large data sets than QP.

Methods computing distance between bags also become impractical as the data set size increases (Amores, 2013). Obtaining the distance between two bags often means computing

[5] https://github.com/macarbonneau/MILSurvey

the distance between each pair of instances, which implies a classification cost of $\mathcal{O}(b^2m^2d)$, where b is the number of bags, m is the average number of instances per bag and d the dimensionality of the data. This becomes to $\mathcal{O}(b^2m^3d)$ when using the earth mover's distance (EMD) to compare the distribution in the two bags. Moreover, these methods must store the entire data set in memory which can also be problematic. To avoid these costs when comparing bags, it is preferable to use bag embedding techniques (Wei *et al.*, 2014). Representing bags as a single feature vector greatly reduces the number of training examples fed to the classifier, when compared to instance based methods. However, not all embedding methods possess the same scalability. For instance, methods representing bags as distance to instance prototypes (e.g. MILES (Chen *et al.*, 2006)) or other bags (Cheplygina *et al.*, 2015c) can produce very high dimensional representation with large data sets (Fu *et al.*, 2011). This can be avoided altogether by representing bags using a vocabulary-like encoding as proposed in (Amores, 2010; Wei *et al.*, 2014). In (Ping *et al.*, 2011; Xu *et al.*, 2017), hash functions have been used to accelerate the bag encoding process. Alternatively, bags can be represented by statistics on the instance as done in the Statistic Kernel (STK) (Gärtner *et al.*, 2002).

While embedding methods decrease the computational cost, they generally do not allow for instance classification. In that case some methods have been proposed to reduce the data set size using instance selection. For example, (Yuan *et al.*, 2014) uses instance selection algorithms inspired by the immune system to reduce the size of the data set before using MIL learning algorithms. MILIS (Fu *et al.*, 2011) has been proposed to reduce the complexity of MILES by selecting only one instance per bag instead of using a 1-norm SVM to perform the selection of prototypes.

Finally, parallelization can be employed to reduce computation time, like in (Cano *et al.*, 2015), where a parallelized version of the G3P-MI (Zafra & Ventura, 2010) algorithm have been proposed to leverage the power of GPUs, and thus deal with large quantities of data.

1.7.5 Future Direction

Based on the literature review of this survey, we identify several MIL topics that are interesting avenues for future research.

First, tasks like regression and clustering are not extensively studied when compared to classification. This might be because there are less applications for these tasks, and because there are no publicly available data sets. A good place to start exploration on MIL regression could be in affective computing applications, where the objective is to quantify abstract concepts, such as emotions and personalities. In these applications, real-valued labels express the appreciation of human judges for speech or video sequences (bags). The sequences are represented by an ensemble of observations (instances), and it is unclear which observation contributed to the appreciation level. In this light, these problems perfectly fit in the MIL framework. Better regression algorithms would also be useful in CAD to assess the progression stage of a pathology instead of only classifying subjects as diseased or healthy.

Also, it is only fairly recent that the difference between instance and bag classification is throughly investigated. It is demonstrated in (Doran & Ray, 2014a; Vanwinckelen *et al.*, 2015), in Section 1.4.1 and our experiments that these tasks are different. It is showed in this paper and (Carbonneau *et al.*, 2016d) that many instance-space methods proposed for bag classification are sub-optimal for instance classification. There is a need for MIL algorithms primarily addressing instance classification, instead of performing it as a side feature. Based on the results Section 1.6.2 approaches discarding or only minimally using the bag arrangement information appears to be better suited for this task. We believe that this bag arrangement could be better leveraged than how it is done by existing methods, which often seek to maximize bag-level accuracy. To further stimulate research on this topic, more instance-annotated MIL data sets are needed.

In some applications, the training data contains only positive and unlabeled data. For example, in recommender systems, the history of a user contains a list of consulted products that can be modeled as bags. If the user bought a product, it is considered as a positive bag. The

other consulted products may or may not be interesting to the customer and therefore remain unlabeled. This type of problem is well studied in single-instance learning (Zhang & Zuo, 2008), but requires more exploration in the MIL context. As explained before, and observed in the experiments, many MIL methods performance depends on the characterization of the negative distribution and the correctness of bag labels to identify positive concepts. In this case, learning from positive and unlabeled bags becomes a difficult problem for MIL. So far, only a handful of papers are dedicated to this subject (Wu *et al.*, 2014d; Bao *et al.*, 2017; Wu *et al.*, 2017).

While tasks outside bag classification would benefit from more exploration, there are also problem characteristics that necessitate the attention of the MIL community. For instance, intra-bag similarities have never been identified as a challenge, and thus, directly addressed. It could be beneficial to perform some sort of normalization or calibration in each bag to remove what is common to each instance and specific to the bag. In computer vision, this is usually done in a preliminary normalizing step. However, in other tasks such as molecule classification, this type of procedure could be helpful. For example, in the Musk data, the instances in the bag are conformations of the same molecule. Discarding the information related the "base" shape of the molecule could help to infer what more subtle particularity of the configurations is responsible for the effect when comparing to other molecules.

There are only a few methods that leverage the structure in bags. This is an important topic that has been addressed in some BoW methods, but was never thoroughly studied in other types of MIL methods, except for some methods using graphs (Zhou *et al.*, 2009; Yan *et al.*, 2016; Zhang *et al.*, 2011b; Wu *et al.*, 2014b; Mcgovern & Jensen, 2003). Some of these methods represent similarities between instances or represent whole bag as graphs. Methods that create an intermediate graph representation in which some instances are grouped in sub-graphs could be an interesting way to leverage the inner structure of bags. In that case, the witness would correspond to an ordered arrangement of instances. With this type of representation, complex objects could be identified more reliably in complex environments.

In many problems, the numbers of negative and positive instances are severely imbalanced, and yet, the existing learning methods for imbalanced data set have not studied extensively in MIL. There exist many methods to deal with imbalanced data (Branco *et al.*, 2016). There are external methods like SMOTE (Chawla *et al.*, 2002) and RUSBoost (Seiffert *et al.*, 2010) that necessitate accurate labels to perform over or under sampling. To be adapted to MIL these methods could use some kind of probabilistic label function. Internal methods (Imam *et al.*, 2006; Veropoulos *et al.*, 1999) adjust the misclassification cost independently for each class. These schemes could be used in algorithms such as mi-SVM which require the training of an SVM with high class imbalance when the WR is low. Class imbalance has also been identified in (Herrera *et al.*, 2016a) as an important topic for future research.

When working with MIL, one must deal with uncertainty. It would be beneficial in many applications to use active learning to train better classifiers by querying humans about most uncertain parts of the feature space. For example, in CAD, after preliminary image classification, the algorithm would determine which are the most critical instances and prompt the clinician to provide a label. These critical instances would be the most ambiguous or the ones that would most help the classifier. This would necessitate research to assert degrees of confidence in parts of feature space. Existing literature on this subject is rather limited (Settles *et al.*, 2008; Meessen *et al.*, 2007; Melendez *et al.*, 2016b; Zhang *et al.*, 2010). Alternatively, the algorithm should be able to evaluate the information gain that each instance label would provide. As a related topic, new methods should be proposed to incorporate knowledge from external and reliable sources. Intuitively, the information obtained with strong labels should have more importance in the MIL algorithm's learning and decision process than instance with weak labels.

Except for a few papers, MIL methods always focus on classification/regression, and features are considered as immutable parameters of the problem. Recently, methods for representation learning (Bengio *et al.*, 2013) have gained in popularity because they usually yield a high level of accuracy. Some of these methods learn features in a supervised manner to obtain a more discriminative representation (Mairal *et al.*, 2008), or, in deep learning, a supervised training

phase is often used to fine tune the features learned in an unsupervised manner (Larochelle *et al.*, 2009). This cannot be done directly in MIL because of the uncertainty on the labels. The adaptation of discriminative feature learning methods would be beneficial to MIL. Also, it has be shown that mid-level representation help to bridge the semantic gap between low-level features and concepts (Hauptmann *et al.*, 2007; Li *et al.*, 2010; Sadanand & Corso, 2012). These methods obtain a mid-level representation using supervised learning on images or videos annotated with bounding boxes. Learning techniques for these mid-level representations should also be proposed for MIL. This is an area where multiple instance clustering would be useful. There are already a few papers on this promising subject (Zhu *et al.*, 2015, 2013). However, there are still a lot of open questions and limitations to overcome, such as dealing with multiple objects in a single image or the dependency to a saliency detector.

In some applications, like emotion or complex event recognition from videos, objects are represented using different modalities. For example, the voice and facial expression of a subject can be used to analyze its behavior or emotional state (Ringeval *et al.*, 2013). Alternatively, events in videos can be represented, among others, by frame, texture and motion descriptors (Merler *et al.*, 2012; Tang *et al.*, 2013). In both cases, a video sequence is represented by a feature vector collection corresponding to a bag. The difference with existing MIL problems is that these instances belong to different feature spaces. This is analogous to multi-view MIL which has been studied in a few papers (Wu *et al.*, 2013, 2014c,a; Nguyen *et al.*, 2013). This interesting problem necessitates more research from the MIL community, and will find applications in areas, such as multimedia analysis or problems related to the Internet-of-things, which necessitate the fusion of diverse information sources. By their nature these applications imply large quantity of data, and thus MIL would allow exploiting all this information and reduce the burden of annotation. Several fusion strategies should be explored. Instances could be mapped to the same semantic space to be compared directly, graph model could be used to aggregate several heterogeneous descriptors or instances could be combined in pairs to create new spaces for comparison similarly to (Daumé III, 2009).

1.8 Conclusion

In this paper, the characteristics and challenges of MIL problems were surveyed with applications in mind. We identified four types of characteristics which define MIL problems and dictate the behavior of MIL algorithms on data sets. It is an important topic in MIL because a better knowledge of these MIL characteristics helps interpreting experiments results and may lead to the proposal of improved methods in the future.

We conducted experiments using 16 methods which represent a broad spectrum of approaches. The experiments showed that these characteristics have an important impact on performance. It was also shown that each method behaves differently given the problem characteristics. Therefore, careful characterization of problems should not be neglected when experimenting and proposing new methods. More specific conclusions have also been drawn from experiments:

- For instance classification tasks, when the WR is relatively high, there is no need for MIL algorithms. The problem can be cast as a regular supervised problem with one-sided noise;
- For instance classification tasks, the best approaches do not use use bag information (or only very lightly). Also, methods optimized using bag classification accuracy as an objective have a higher false negative rate (as the WR increases), which limits their performance for this task;
- Bag-space methods and methods assuming instances inherit their bag label yield better classification performance especially when the WR is high;
- Bag-space methods are more robust than instance-space methods in problems where the negative distribution cannot be completely represented by the training data. This was particularly true when using the minimal Hausdorff distance;
- Embedding-space methods are robust to label noise, while instance-space methods are not;

- Measuring performance only in terms of AUC is misleading. Some algorithms learn an accurate score function, but fail to optimize the decision threshold used to obtain hard labels, and thus, yield low accuracy.

After observing how problem characteristics impact MIL algorithms, we discussed the necessity of using more benchmark data sets than the Musks and Tiger, Elephant and Fox data sets to compare proposed MIL algorithms. It became evident that appropriate benchmark data sets should be selected based on the characteristics of the problem to be solved. We then identified promising research avenues to explore in MIL. For example, we found that only few papers address MIL regression and clustering, which is useful in emerging applications such as affective computing. Also, more methods leveraging structure among instances should be proposed. These methods are in high demand in the era of the Internet of things, where large quantities of time series data are generated. Finally, methods dealing efficiently with large amount of data, multiple modalities and class imbalance require further investigation.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

CHAPTER 2

ROBUST MULTIPLE-INSTANCE LEARNING ENSEMBLES USING RANDOM SUBSPACE INSTANCE SELECTION

Marc-André Carbonneau[1,2], Eric Granger[1], Alexandre J. Raymond[2], Ghyslain Gagnon[2]

[1] Laboratory for Imagery, Vision and Artificial Intelligence,
École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] Communications and Microelectronic Integration Laboratory,
École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article published in « Elsevier's Pattern Recognition » in April 2016.

Abstract

Many real-world pattern recognition problems can be modeled using multiple-instance learning (MIL), where instances are grouped into bags, and each bag is assigned a label. State-of-the-art MIL methods provide a high level of performance when strong assumptions are made regarding the underlying data distributions, and the proportion of positive to negative instances in positive bags. In this paper, a new method called Random Subspace Instance Selection (RSIS) is proposed for the robust design of MIL ensembles without any prior assumptions on the data structure and the proportion of instances in bags. First, instance selection probabilities are computed based on training data clustered in random subspaces. A pool of classifiers is then generated using the training subsets created with these selection probabilities. By using RSIS, MIL ensembles are more robust to many data distributions and noise, and are not adversely affected by the proportion of positive instances in positive bags because training instances are repeatedly selected in a probabilistic manner. Moreover, RSIS also allows the identification of positive instances on an individual basis, as required in many practical applications. Results obtained with several real-world and synthetic databases show the robustness of MIL ensembles designed with the proposed RSIS method over a range of witness rates, noisy features and data distributions compared to reference methods in the literature.

2.1 Introduction

Multiple-instance learning (MIL) is a form of weakly-supervised learning (Ikeuchi, 2014), where data *instances* are grouped into *bags*. A label is not provided for each instance, but for a whole bag. Typically, a negative bag contains only negative instances, while positive bags contain instances from both classes (Dietterich *et al.*, 1997).

Since the first formulations of the MIL problem (Dietterich *et al.*, 1997; Keeler *et al.*, 1990) many solutions have been proposed. In many cases, MIL algorithms were developed with a specific application in mind. For instance, Diettrich (Dietterich *et al.*, 1997) proposed Axis Parallel Rectangle (APR) to solve a molecule classification problem. Later, many methods were proposed to solve image categorization (Andrews *et al.*, 2002; Chen & Wang, 2004; Chen *et al.*, 2006; Fu *et al.*, 2011; Rahmani & Goldman, 2006), web mining (Zhou *et al.*, 2005a; Zafra *et al.*, 2007), object and face detection (Viola *et al.*, 2006; Babenko *et al.*, 2011a; Guillaumin *et al.*, 2010; Vijayanarasimhan & Grauman, 2008; Ali & Saenko, 2014) and tracking (Babenko *et al.*, 2011c) problems. While they can achieve a high level of performance in their respective application domains, many of these methods are less efficient over a wide variety of data distributions and pattern classification problems.

For instance, many methods rely on the assumption that the proportion of positive instances in positive bags, hereafter called *witness rate*, is high. Sometimes, these methods implicitly assume that all instances in a positive bag are positive. This is the case for methods such as APR (Dietterich *et al.*, 1997), Citation-kNN (Wang & Zucker, 2000) and diverse density-based (DD) methods (Chen & Wang, 2004; Chen *et al.*, 2006; Maron & Lozano-Pérez, 1998; Zhang & Goldman, 2001). This assumption is also made in the initialization of the optimization process in mi-SVM and MI-SVM (Andrews *et al.*, 2002). Other methods assume a high witness rate by representing bags as the average of the instances it contains, as in MI-Kernel (Gärtner *et al.*, 2002) and MIBoosting (Xu & Frank, 2004). The performance of all these methods decreases when the high witness rate assumption is not verified, which limits the applicability of MIL methods to many problems. For instance, until recently, object identification systems

were limited to problems where instances represent slight translational and scale uncertainties around localization bounding boxes (Ali & Saenko, 2014).

To deal with lower witness rates, Gehler and Chapelle (Gehler & Chapelle, 2007) applied deterministic annealing to an SVM-based MIL algorithm. Bunescu and Mooney (Bunescu & Mooney, 2007b) enforced the constraint that positive bags contain at least one positive instance in their SVM formulation. Both obtained good results with lower witness rates, but observed performance degradation with higher witness rates. SVR-SVM (Li & Sminchisescu, 2010) and the γ-rule (Li *et al.*, 2013) have been proposed to overcome these problems by estimating the witness rate and then using it as a system parameter. These techniques provide a high level of performance over a range of high and low witness rates, yet, the witness rate is assumed to be constant across all bags. This assumption proves to be problematic in some applications, such as image categorization (Zhang *et al.*, 2002), where images are segmented and features are extracted from the different segments (Andrews *et al.*, 2002; Chen & Wang, 2004). The image corresponds to a bag, while each segment is an instance. Depending on the visual complexity of the image, a different proportion of target and non-target segments will be obtained. Therefore, the witness rate of a bag depends on the image content, and is likely to vary from one bag to another.

Another challenge of MIL problems is the fact that the shape of positive and negative distributions affect the performance of some algorithms. For instance, some methods such as APR (Dietterich *et al.*, 1997) are not designed to deal with multi-modal distributions where instances are grouped in distinct clusters. Methods based on DD (Chen & Wang, 2004; Chen *et al.*, 2006; Maron & Lozano-Pérez, 1998; Zhang & Goldman, 2001) assume that positive instances form a compact cluster (Fu *et al.*, 2011). In MILIS (Fu *et al.*, 2011), the negative distribution is modeled with Gaussian kernels, which can be difficult when the quantity of data available is limited. On the other hand, in Citation-kNN (Wang & Zucker, 2000) the presence of compact data cluster in the negative distribution increases the probability of misclassification.

Finally, some methods classify bags as a whole instead of trying to label each instance individually. Some of these methods (Wang & Zucker, 2000; Gärtner *et al.*, 2002; Cheplygina *et al.*, 2015c; Zhou *et al.*, 2009) use different types of bag distance measure, while others embed bags using distance to a set of prototypes (Chen *et al.*, 2006; Fu *et al.*, 2011; Chen & Wang, 2004), vocabulary (Amores, 2010) and sparse coding (Song *et al.*, 2013). Bag-level classification approaches cannot identify instances individually, which is necessary in certain applications such as object detection and tracking in images or videos. Moreover, by considering bags as a whole, the performance of these methods often decreases in problems where the witness rate is low.

To address these limitations, this paper proposes a new ensemble-based method for MIL called Random Subspace Instance Selection (RSIS). Classifier ensembles are generally known to provide accurate and robust classification systems when data is limited (Kuncheva, 2004). The key feature of RSIS is that it constructs classifier ensembles based on a probabilistic identification of positive instances. The proposed method allows to classify instances individually and does not rely on a specific witness rate or specific type of data distribution. It can therefore be applied in a wide variety of context.

In the proposed method, the training data is projected onto several random subspaces before being clustered. The proportion of instances from positive and negative bags is computed for every cluster. Based on these bag proportions, a *positivity score* is computed for every instance in the data set. These scores are later converted into selection probabilities, and used to select diverse training sets to generate base classifiers in the ensemble. The general intuition for RSIS is that it is easier to identify positive instance clusters while only considering a discriminant subset of features. The optimal feature subset to represent a given concept is unknown, and may vary from one concept to another. However, if a data set is projected into all possible subspaces, instances from the same concept are more likely to be grouped together than with the other instances.

The RSIS method allows to design MIL ensembles that are robust to various witness rate, because each time one of the classifiers in the ensemble is trained, only one instance is used from each bag. The instances are drawn based on their probability of being positive. If the witness rate is low and only one instance is likely to be positive, this instance will be the only one selected. In contrast, if many instances appear to be positive, each instance will have a similar probability of being selected, and thus being used as a training instance in one or another classifier. Since selection probabilities are computed for each bag separately, the witness rate does not have to be constant across all bags. Moreover, by clustering the data in many different subspaces, RSIS can inherently uncovers multiple underlying concepts in the data distributions. This makes the algorithm resistant to multi-modal distributions of various shapes, and robust to noisy or irrelevant features.

In this paper, the performance of MIL ensembles designed using RSIS is compared to several methods in the literature using benchmark data sets. Further experiments are performed on synthetic data sets to study the algorithm's tolerance to various multi-modal distributions, witness rate and irrelevant features. Five well-known baseline methods, APR (Dietterich *et al.*, 1997), Citation-kNN (Wang & Zucker, 2000), mi-SVM (Andrews *et al.*, 2002), AL-SVM (Gehler & Chapelle, 2007) and CCE (Zhou & Zhang, 2007) are also used for comparison. Finally, the sensitivity of the proposed approach to internal parameters is also characterized experimentally, and some general guidelines for parameter selection are provided.

The remainder of this paper is organized as follows. The MIL problem is formalized and state-of-the-art techniques are reviewed in Section 2.2. Then, in Section 2.3, the proposed RSIS algorithm is described. Section 2.4 presents the experimental methodology. Section 2.5 presents robustness experiments on synthetic data, while Sections 2.6 and 2.7 present experimental results on benchmark data sets, and experiments on parameter sensitivity respectively. Time complexity is discussed in Section 2.8.

2.2 Multiple Instance Learning

Let $\mathcal{B} = \left\{B^1, B^2, ..., B^Z\right\}$ be a set composed of Z bags[1]. Each bag B^i corresponds to a positive or negative label $L^i \in \{-1, +1\}$ in the set $\mathcal{L} = \left\{L^1, L^2, ..., L^Z\right\}$, and contains N^i feature vectors: $B^i = \left\{\mathbf{x}^i_1, \mathbf{x}^i_2, ..., \mathbf{x}^i_{N^i}\right\}$ where $\mathbf{x}^i_j = (x^i_{j1}, x^i_{j2}, ..., x^i_{jd}) \in \mathbb{R}^d$. Each of these feature vector instances corresponds to a positive or negative label in the set $Y^i = \left\{y^i_1, y^i_2, ..., y^i_{N^i}\right\}$, where $y^i_j \in \{-1, +1\}$. Instance labels are unknown in positive bags, but are assumed negative in negative bags. A bag is labeled positive if at least one instance contained in the bag is labeled positive (Dietterich *et al.*, 1997):

$$L^i = \begin{cases} +1 & \text{if} \quad \exists y \in Y^i : \; y^i_j = +1; \\ -1 & \text{if} \quad \forall y \in Y^i : \; y^i_j = -1. \end{cases} \tag{2.1}$$

Many methods have been proposed over the years to address MIL problems in a variety of domains. An overview of these methods and a review of the MIL assumptions can be found in recent surveys by Amores (Amores, 2013) and Foulds and Frank (Foulds & Frank, 2010). In the taxonomy proposed by Amores, (Amores, 2013) MIL methods are divided in three categories, based on how bags are represented. A first corpus of methods operates at the instance level. Each instance is classified individually, and scores are aggregated to label bags. The two other types of method operate on the bag level. In one case, bags are mapped to a vector representation, which reformulate the MIL problem as a standard supervised classification problem, while in the other case, distance metrics are proposed to compare whole bags.

The proposed method falls in the instance-level category. When operating at this level, it is not only possible to categorize bags but also to identify positive instances in bags individually. This is necessary in some application such as object detection and tracking applications (Chen *et al.*, 2006; Viola *et al.*, 2006; Babenko *et al.*, 2011a; Guillaumin *et al.*, 2010; Vijayanarasimhan & Grauman, 2008; Ali & Saenko, 2014; Babenko *et al.*, 2011c). There exists

[1] Throughout this paper, upper indexes are used to denote bags, while lower indexes designate instances. For the sake of clarity, when unnecessary, these indexes are omitted.

many instance-level techniques in the literature, starting with APR, proposed as early as 1997 by Diettrich et al. (Dietterich *et al.*, 1997). In this method, an hyper-rectangle is expanded and shrunk to maximize the number of instances from positive bags, while minimizing the number of instances from negative ones. Instances falling inside the hyper-rectangle are considered positive, while others are labeled negative. APR considers all instances in positive bags to be positive, and, thus, assumes a high witness rate. Also, the use of an hyper-rectangle as a single classification region implies the assumption that positive instances come from a single cluster in space.

Maron and Lorenzo-Pérez proposed to use the diverse density (DD) measure (Maron & Lozano-Pérez, 1998). The DD of a location in feature space is high if its neighborhood contains many instances from different positive bags and few from negative bags. Later, with EM-DD, Zhang and Goldman (Zhang & Goldman, 2001) proposed to use the Expectation-Maximization algorithm to search for the maxima of the DD function. DD-based methods work under the assumption that the positive data comes from a compact clusters in feature space (Fu *et al.*, 2011), which limits their applicability in many problems. Also, DD and EM-DD performance decreases with number of relevant features (Zhang & Goldman, 2001).

In some methods bags are represented by averaging the instances they contain. In MI-Kernel (Gärtner *et al.*, 2002), a bag is summarized by a normalized sum of the instances it contains. In MILBoost (Xu & Frank, 2004) the probability of a bag being positive is obtained by averaging the probabilities of each instance it contains. By pooling all instances together, these methods assume a high witness rate.

Many max-margin classifiers were proposed for MIL problems. These methods were recently surveyed and analyzed by Doran and Ray (Doran & Ray, 2014a). Andrews et al. (Andrews *et al.*, 2002) were among the firsts to extend SVMs to solve MIL problems. Two algorithms were proposed: mi-SVM and MI-SVM. In mi-SVM, the margin is maximized jointly over instance label assignations and a discriminant function. Every instance found in a positive bag is initialized as positive. The SVM is first trained based on these assignments. The resulting

classifier is then used on the same training data to update the instance labels. Next, the SVM is trained based on the new label assignments, and so forth. The second algorithm, MI-SVM, focuses on maximizing the margin over the bags instead of instances by choosing a single instance to represent bags. MICA works similarly but selects a convex combination of witnesses to represent bags (Mangasarian & Wild, 2008). By initializing all instance labels in positive bags as positive, these methods rely on the assumption that the witness rate is high.

To deal with lower witness rates, Gehler and Chapelle (Gehler & Chapelle, 2007) applied deterministic annealing to the aforementioned SVM-based MIL algorithms. With Sparse-MIL, Bunescu and Mooney (Bunescu & Mooney, 2007b) proposed to enforce the constraint that there is at least one positive instance in each positive bag in a transductive SVM formulation. Both methods obtain a high level of performance at low witness rates, but observe performance degradation at higher witness rates.

To address the performance dependency to specific witness rates, Li and Sminchisescu proposed SVR-SVM (Li & Sminchisescu, 2010). In SVR-SVM, the MIL problem is formulated as a convex joint estimation of the likelihood ratio function and the likelihood ratio values on training instances. They obtained high level of performance at high and low witness rate, but assumed the witness is constant across all bags.

Chen and Wang (Chen & Wang, 2004) used DD and SVM to embed and classify bags. DD-SVM selects multiple instance prototype corresponding to local maxima of the DD response function. Bags are represented by distance from these prototypes. This idea was later used in MILES (Chen *et al.*, 2006), except that instances from the training set are used, instead of prototype, to embed bags. While yielding high level of performance, the method does not scale well to large problems, since the dimension of bag feature vectors depends on the number of training instances in the data set (Fu *et al.*, 2011). Fu et al. (Fu *et al.*, 2011) proposed MILIS to minimize this problem, with an initial selection of the prototype instances via several runs of EM-DD.

Zhou and Zhang proposed CCE (Zhou & Zhang, 2007), an algorithm based on clustering and classifier ensembles. Training data is clustered, and the bags are represented as binary vectors in which each bit corresponds to a cluster. A bit is set to 1 if at least one instance of the bag is attributed to its corresponding cluster. To design the ensemble, several clusterings are performed and a classifier is trained using each different data representation. This method represents whole bags based on clustering results, while with ensembles created with RSIS classify instances individually in the original feature space.

Other ensemble methods have been proposed to solve MIL problems. For instance, many authors proposed variations of boosting for object detection (Viola *et al.*, 2006; Ali & Saenko, 2014; Xu & Frank, 2004), while others proposed to combine different classifiers (Zhou & Zhang, 2003). Li et al. proposed the γ-rule for classifier combination in MIL contexts (Li *et al.*, 2013). They assume that instances in data sets can be modeled as a mixture of concept and non-concept distributions. Once estimated, the mixture is used to re-weigh the posteriors of classifiers. In this method, the witness rate is estimated, and is assumed to be constant across all bags.

Some methods, like Citation-kNN (CkNN) proposed by Wang and Zucker (Wang & Zucker, 2000), operate at the bag level. This method is inspired by the notion of citations in research. For a given bag b, the r nearest *references* correspond to the r nearest bags, using the Hausdorff distance. The nearest *citers* are the bags that count b in their c nearest bags. The label of bag b is obtained by a majority vote on the *reference* bags and *citers* bags pooled together. Many other methods use bag distance measures such as the dissimilarity measure (Cheplygina *et al.*, 2015c), or the graph kernels (Zhou *et al.*, 2009).

For most of these methods, strong assumptions have been made implicitly or explicitly regarding the witness rate and the data distribution. When very little is known about the nature of the data and the content of the bags, selecting a robust MIL method can be difficult. The proposed RSIS method presented in Section 2.3 is a general method that allows to design discriminant MIL ensembles without prior assumptions regarding witness rate and data distributions. Classifier ensembles are known to handle complex data structures and to provide better generalization

and accuracy than single classifier systems (Kuncheva, 2004). Moreover, because the proposed method classifies instances individually, it can be used in MIL problems like object tracking and detection for which bag-based methods cannot be used.

2.3 Random Subspace Instance Selection for MIL Ensembles

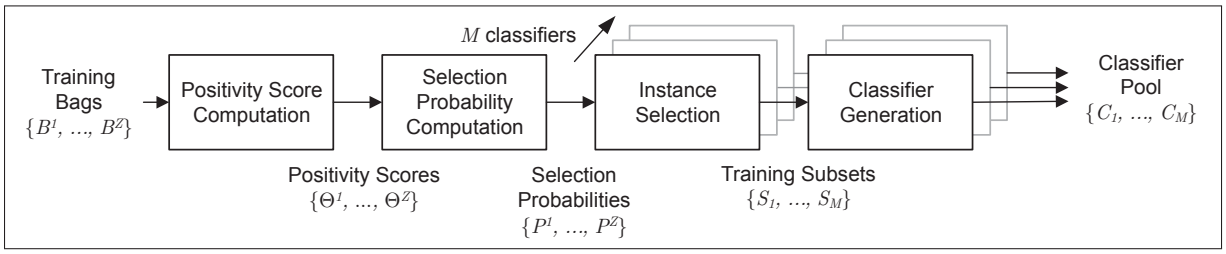


Figure 2.1 MIL ensemble design using the proposed RSIS technique

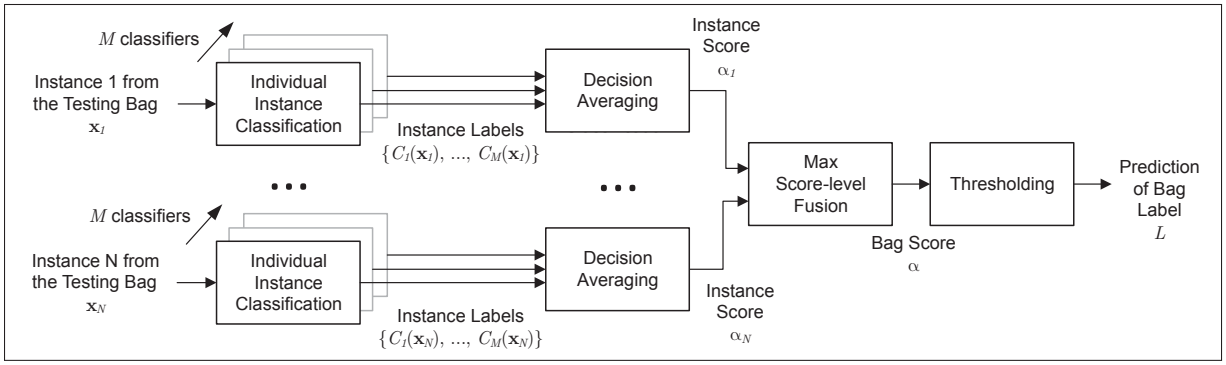


Figure 2.2 Bag label prediction using MIL ensemble

The basic steps of the proposed approach for MIL ensemble design using RSIS are represented in Figure 2.1. At first, each instance receives a *postivity score* based on clustering of data in random subspaces, which indicates the likelihood that an instance is positive. The computation of these scores is described in Section 2.3.1. Given these scores, an instance selection probability distribution is obtained for each bag. To generate a diverse pool of base classifiers, each one is trained on a different subset of the training data, where each subset contains one instance from each positive bags and instances from the negative bags. These instances are randomly

selected based on the previously computed instance selection probability distribution. This process may be viewed as a variation on bagging (Breiman, 1996), with the novelty that subset sampling is guided by the positivity scores. Ensemble design is detailed in Section 2.3.2.

As depicted in Figure 2.2, when an unknown test bag is presented to the system during operation, each classifier predicts a label for each instance. The decisions of the classifiers are averaged to produce a score for each instance. The highest instance score is attributed to the bag, and this bag score is compared to a threshold for final prediction of class label. Bag classification is described in Section 2.3.3.

2.3.1 Positivity Score Computation

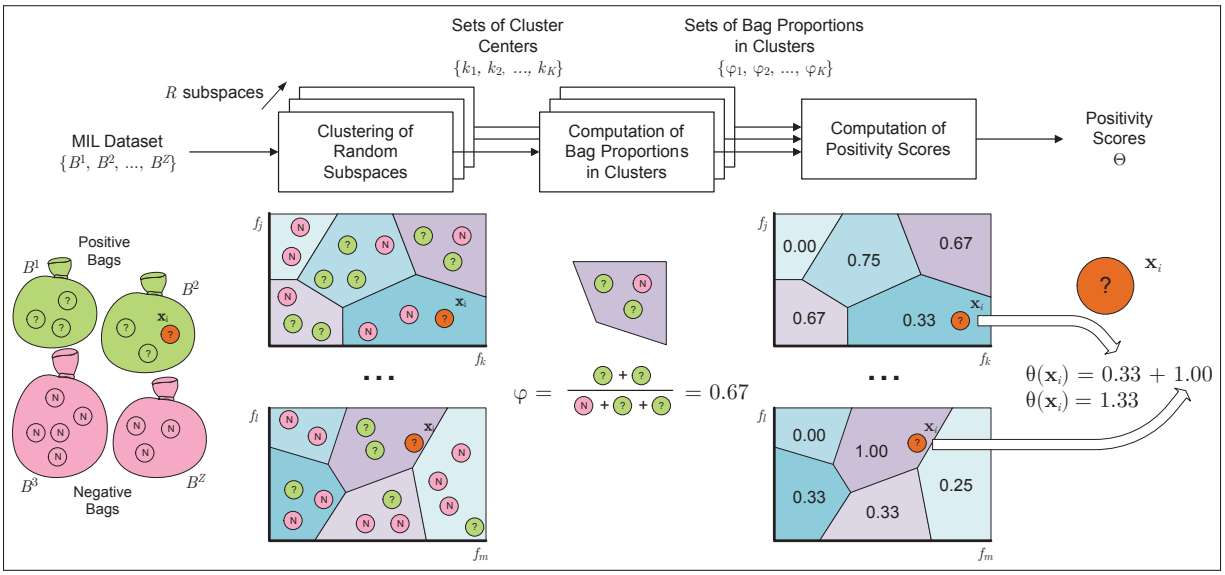


Figure 2.3 Illustration of the pipeline to compute positivity scores with RSIS

The computation of positivity scores is illustrated in Figure 2.3 and summarized in Algorithm 2.1. The first step consists in randomly selecting p features from the complete set of d features to create a subspace $\mathcal{P}$. If $\mathcal{F}$ is the complete space, then $\mathcal{P} \subseteq \mathcal{F}$.

Every instance $\mathbf{x}$ from each bag B^i is projected onto the subspace $\mathcal{P}$. A clustering of this space is then performed. Next, the proportion φ_n of instances belonging to positive bags is computed

for each cluster k_n, where $n = 1, 2, ..., K$:

$$\varphi_n = \frac{\sum_{\forall \mathbf{x}} c(\mathbf{x}^i, n)}{|\mathcal{K}_n|} \in [0..1], \tag{2.2}$$

where:

$$c(\mathbf{x}^i, n) = \begin{cases} 1, & \text{if } \mathbf{x}^i \in \mathcal{K}_n \text{ and } L^i = +1; \\ 0, & \text{otherwise.} \end{cases} . \tag{2.3}$$

In these equations, $\mathcal{K}_n$ is the set of instances belonging to cluster k_n, and $|\mathcal{K}_n|$ is the size of this set.

The complete process of selecting a random subspace, projecting the data into the subspace and clustering the projected data is repeated R times. At the end of repetition $r = 1, 2, ..., R$, each instance $\mathbf{x}$ receives the positive bag proportion $\varphi_n(r)$ of its cluster assignment. The values from all repetitions are summed in order to get a positivity score set $\Theta^i = \left\{\theta_1^i, \theta_2^i, ..., \theta_{N^i}^i\right\}$ in which each value corresponds to an instance in the data set:

$$\theta(\mathbf{x}) = \frac{1}{R} \sum_{r=1}^{R} \sum_{n=1}^{K} \varphi_n(r) \cdot d(\mathbf{x}, n, r), \tag{2.4}$$

where

$$d(\mathbf{x}, n, r) = \begin{cases} 1, & \text{if } \mathbf{x} \in \mathcal{K}_n \text{ at repetion } r; \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

Positivity scores indicate the likelihood that the instances belong to the positive class. In positive bags, these scores indicate the most likely positive instances, while in negative bags, they allow to rank instances according to classification difficulty.

2.3.2 Ensemble Design

Each classifier in the pool $\mathcal{C} = \{C_1, C_2, ..., C_M\}$ maps instances to binary hard labels: $C : \mathbb{R}^d \rightarrow \{0, 1\}$. Each classifier is trained on a different data subset $\mathcal{S}_p$ composed of instances selected

Algorithm 2.1 Computation of positivity $\theta(\mathbf{x})$ score for each instance $\mathbf{x}$

Data: Training set $\mathcal{B}$
Result: Positivity score set Θ

for *r = 1 to R subspaces* **do**
    randomly select a p-feature subspace $\mathcal{P}$;
    project all instances in $\mathcal{B}$ onto subspace $\mathcal{P}$;
    perform clustering of projected data using K cluster centers;
    for *n = 1 to K clusters* **do**
        compute $\varphi_n(r)$ using Eq. (2.2);
    end
end
for $\forall \mathbf{x} \in \mathcal{B}$ **do**
    compute score $\theta(\mathbf{x})$ for using Eq. (2.4);
end
return positivity score set Θ ;

based on the positivity scores Θ^i (see Eq. (2.4) in Section 2.3.1). At this point, the domain of the instances is the entire feature space. In each bag, these scores are converted to selection probabilities by applying a soft-max function on all instances it contains, and one instance $\mathbf{x}_*$ is selected per bag:

$$P(\mathbf{x}_* = \mathbf{x}_k|\Theta) = \frac{e^{\theta_k/T}}{\sum_{j=1}^{N_i} e^{\theta_j/T}}, \tag{2.6}$$

where $T \in \mathbb{R}^+$ is the temperature parameter. The training subset is created by choosing one instance from each bag based on the selection probabilities. The label of the selected instances corresponds to the label of their bags ($y_j^i = L^i$).

Finally, classification performance can be enhanced by adding randomly selected instances from negative bags to the training subsets.

Algorithm 2.2 Generation of classifier pools with the RSIS method

Data: Training set $\mathcal{B} = \left\{B^1, B^2, ..., B^Z\right\}$
Result: Classifier pool $\mathcal{C}$

initialize $\mathcal{C} = \emptyset$;
compute positivity scores (see Algorithm 2.1);
compute selection probabilities $P(\cdot|\Theta)$ using Eq. (2.6);
for $i = 1$ *to* M **do**
    $\mathcal{S} = \emptyset$;
    for $j = 1$ *to* Z **do**
        select one instance $\mathbf{x}_*^j$ using $P(\cdot|\Theta^j)$;
        add it to the training subset $\mathcal{S}$;
    end
    add randomly selected instances from negative bags to $\mathcal{S}$;
    train classifier C_i using $\mathcal{S}$;
    add C_i to pool $\mathcal{C}$;
end
return classifier pool $\mathcal{C}$;

2.3.3 Prediction of Bag Labels

During operation, each unknown test instance is classified individually, and a bag is deemed positive when it contains a positive instance. Formally, the label L of a bag B is given by:

$$L = \begin{cases} +1, & \text{if } \alpha > \beta; \\ -1, & \text{otherwise,} \end{cases} \tag{2.7}$$

where β is a threshold set empirically on validation data, and $\alpha \in [0,1]$ is the averaged outputs of the classifiers for the *most positive* instance in the bag:

$$\alpha = \max_{\mathbf{x} \in B} \left\{ \frac{1}{M} \sum_{j=1}^{M} C_j(\mathbf{x}) \right\}. \tag{2.8}$$

In applications such as tracking and object recognition, labeling bags is not sufficient. The algorithm must identify which instances in the bag are the most likely to be positive. Using the proposed algorithm, this translates to simply ranking and selecting the instance $\hat{\mathbf{x}}$ with the highest score in a positive bag if only one instance is needed:

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in B}{\operatorname{argmax}} \left\{ \frac{1}{M} \sum_{j=1}^{M} C_j(\mathbf{x}) \right\}. \tag{2.9}$$

In applications where more than one instance needs to be selected, the threshold β is applied to each instance, as performed for bags in Eq. (2.7).

2.3.4 Why it Works

In essence, ensemble design with RSIS is akin to Bagging (Breiman, 1996). There are theoretical and experimental evidences that Bagging pushes unstable classification procedures, such as classification trees and neural nets, towards optimality in prediction (Breiman, 1996). Ensembles created through some Bagging procedure consist of classifiers trained with different subsets of instances. In supervised learning problems, any instances can be randomly selected. However, in MIL problems, blind selection of the instances may result in poor classifier and ensemble performance. This is because negative instances may be used as positive instances, which would introduce noise into the training data. For example, if the witness rate is as high as 50%, half of the training instances of a class are incorrectly labeled. Integrating a positive instance identification and selection mechanism into the ensemble design procedure, is a key idea of the proposed RSIS algorithm. The remainder of the section presents an analysis of the positive instance identification process in RSIS.

Let us consider data with two underlying concepts (one positive and one negative) that do not overlap in the input feature space. In an ideal case, the clustering process would result in two distinct clusters, each corresponding to a concept. In MIL problems, the cluster corresponding to the negative concept will contain instances from the positive and negative bags, where the proportion of instances from positive bags (see Eq. 2.2) depends on the witness rate and the

proportion of positive bags in the data set:

$$\varphi- = (1 - WR)\frac{\|\mathcal{B}^+\|}{\|\mathcal{B}\|} \tag{2.10}$$

Since positive instances cannot be found in negatives bags, the proportion of instances from positive bags in the positive cluster will be 1 ($\varphi_+ = 1$). In this simple example with ideal clustering, the positivity score of a negative instance (see Eq. 2.4) is given by $\theta(\mathbf{x}_-) = \varphi_- < 1$, while the positivity score of a positive instance is $\theta(\mathbf{x}_+) = 1$, therefore $\theta(\mathbf{x}_+) > \theta(\mathbf{x}_-)$. Data subsets used to train the classifiers are constructed based on these scores. By using a very low temperature parameter ($T \to 0$) in Eq. 2.6, all of the instances selected as positive example will necessarily belong to the positive concept. Furthermore, the negative instances belonging to a positive bag will not be selected. In this ideal case, the value of the witness rate and the proportion of positive bags are of no consequence for positive instance identification.

The assumptions made in the previous example rarely hold in practice. First of all, the result of clustering algorithm is rarely perfect, and the data is not always grouped in distinct clusters. Assuming negative instances of the negative and positive bags come from the same distribution, the worst case clustering would equally distribute the real positive instances between all clusters. In that case, if the data set size tends to infinity, in all clusters, the proportion of instances from positive bags is given by:

$$\varphi\Big|_{Z\to\infty} = \frac{\|\mathcal{B}^+\|}{\|\mathcal{B}\|} \tag{2.11}$$

In this worst case clustering result, the contribution to positivity scores is the same for all instances. Thus, this has no impact on the instance selection probabilities, except for the optimal temperature setting. However, if a clustering happens to group positive instances together, the proportion of instances from positive bags (φ) of each cluster may improve discrimination between positive and negative instances. In RSIS, the data is projected in a number of subspace, and then clustered. Thus, different clustering results are obtained, which are either informative or at worst, do not provide useful information. Thus, as the number of clustered random

subspace increases, the positive instances tend to be identified more accurately. This is observed in results of Section 2.7 concerning parameter sensitivity. In Figure 2.8 (c), one can see performances increase (or remain stable) as the number of generated subspaces increases.

2.4 Experimental Setup

Three different experiments were conducted to assess RSIS performance. In the first experiment, MIL ensembles designed with RSIS are compared to five well-known reference MIL classification methods on synthetic data sets. The experiment is designed to measure the algorithms robustness to various witness rates, data distributions and noisy features. In the second experiment, an ensemble based on RSIS is compared to 29 other state-of-the-art MIL methods on real-world benchmark data sets: the two Musk data sets (Dietterich *et al.*, 1997) and the Tiger, Elephant and Fox data sets (Andrews *et al.*, 2002). Finally, the third experiment studies the impact of RSIS parameters on the MIL ensemble performance.

2.4.1 Data sets

Drug Activity Prediction

The Musk data sets are the most widely used benchmarks for MIL classifier performance evaluation. These data sets were introduced by Dietterich et al.(Dietterich *et al.*, 1997) and are both publicly available from the UCI Machine Learning repository[2]. In this data set, each bag corresponds to a type of molecule, and each instance corresponds to a low-energy conformation of this molecule. The task consists in determining if a molecule is musky or not. For the same molecule, not all conformations are musky, hence comes the MIL problem formulation. Each molecule conformation is described by a 166-dimensional vector. The second data set contains many more instances, mostly negative. Table 2.1 summarizes the two data sets and Table 2.2 reports their estimated WR.

[2] http://archive.ics.uci.edu/ml/

Table 2.1 Properties of the benchmark data sets

Data set	+ Bags	- Bags	Instances	Features	Instances per Bags		
					Min.	Max.	Avg.
Musk1	47	45	476	166	2	40	5
Musk2	39	63	6598	166	1	1044	65
Tiger	100	100	1220	230	1	13	6
Fox	100	100	1302	230	2	13	8
Elephant	100	100	1391	230	2	13	7
Newsgroups	50	50	4006	200	18	65	40

Table 2.2 Estimated WR of the benchmark data sets

Data set	Estimated Witness Rate		
	(Li *et al.*, 2013)	(Li & Sminchisescu, 2010)	(Gehler & Chapelle, 2007)
Musk1	0.82	1.00	1.00
Musk2	0.77	0.90	0.28
Tiger	0.51	0.43	0.60
Fox	0.88	1.00	0.71
Elephant	0.80	0.38	0.58

Tiger, Elephant and Fox:

These three data sets come from the COREL data set (Andrews *et al.*, 2002). The bags in these data sets correspond to animal images. In each data set, there are 100 images of a target animal and 100 images of other random animals. An image corresponds to a bag and the segments in the image are instances. Each instance is described by a 230-dimensional feature vector containing shape, color and texture information. The data set is also publicly available[3] and summarized in Table 2.1.

Some papers (Li & Sminchisescu, 2010; Gehler & Chapelle, 2007; Li *et al.*, 2013) include an estimation of the witness rate for the most popular benchmark data sets. These estimations are reported in Table 2.1, and suggest that, in most of these data sets, a large portion of instances in positive bags are positive. This biases results towards methods that classify bags as a whole instead of individual instances (Li *et al.*, 2013). Also, some methods need a high witness rate

[3] http://www.miproblems.org/mi-learning/

to perform well. In order to assess the performance of the proposed RSIS technique with a low witness rate, the Newsgroup benchmark data set [28] is also used as a benchmark. Finally, we created a new synthetic data set allowing control over witness rate, shape of the data distribution and the proportion of noisy features.

Newsgroups

This set was derived by Settles et al. (Settles *et al.*, 2008) from the *20 Newsgroups* (Lang, 1995) data set corpus. The set contains posts from newsgroups on various subjects. Each bag contains 50 posts from the 20 news categories. In positive bags, 3% of posts belongs to the target class while the other posts are uniformly drawn from all other classes. Each post is represented by 200 TFIDF features. Because of its low witness rate, the data set has been used to highlight the insensitivity to witness rate of the SVR-SVM (Li & Sminchisescu, 2010) method. The data set is publicly available from the same site as the Tiger, Elephant and Fox data sets. The characteristics of the Newsgroups data set are summarized in Table 2.1. The numbers reported are the average value of all 20 data sets.

Synthetic Data

In this data set, different configurations are proposed to assess the performance of the algorithms under different situations. Several parameter configurations are produced with various data distributions, witness rates, number of concepts and number of irrelevant features. The data set is made available publicly[4].

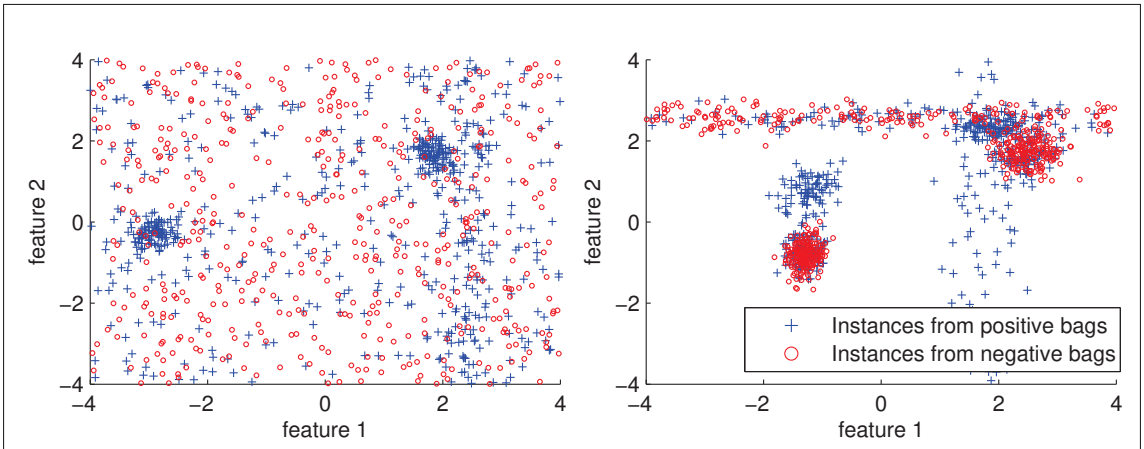
The positive instances are drawn from the concept distribution, while negative instances are drawn either from the uniform distribution $\mathcal{U}(-4,4)$ or from a negative concept distribution. Concept distributions are multivariate Gaussians distributions $\mathcal{G}(\mu,\sigma)$. The values of μ are drawn from $\mathcal{U}(-3,3)$. The covariance matrix (σ) is a randomly generated semi-definite positive matrix in which the diagonal values are scaled to $]0,0.1]$.

[4] http://www.etsmtl.ca/Professeurs/ggagnon/Projects/ai-MIL

In order to model irrelevant features in the data, in each concept, some features are drawn from the uniform distribution instead of the multivariate Gaussian distribution. The number of irrelevant features is controlled by the irrelevant feature proportion (IFP) parameter. For each parameter configuration, the data set is generated 5 times to get results that are more significant. The 10-fold CV procedure is repeated 10 times on each of the 5 generated data sets.

Table 2.3 Default parameters of synthetic data sets

					Instances per Bags		
+ Bags	**- Bags**	**Features**	**IFP**	**Concepts**	**Witness Rate**	**Min.**	**Max.**
100	100	25	0.1	3	0.5	1	50



a) Negative instances from uniform distribution b) Negative instances from concept distribution

Figure 2.4 Example distribution from the synthetic data set. In both a) and b), 2D samples were randomly generated. In a), negative instances are sampled from an uniform distribution, while in b), positive and negative instances are sampled from clustered distributions. Each cluster represents a concept. Markers correspond to bag labels

Examples of 2D data distributions are given in Figure 2.4. In each distribution, one of the features of a concept is irrelevant, which yields the line-shaped cluster. The negative instance distribution is uniform in (a), while negative instances are grouped in Gaussian clusters in (b).

2.4.2 Protocol and Performance Metrics

Experiments were conducted using nested cross-validation (CV) (Stone, 1974) where an inner CV loop is used to select the model parameters, while the outer CV loop is used to estimate the algorithm performance. Both the inner and outer CV loop use 10 folds. At each iteration of the outer loop, a fold is reserved for testing, and model selection is performed via CV grid search on the remaining parts. The best performing configuration is selected by averaging the results obtained on each fold of inner loop CV process. The algorithm is then retrained with the best configuration using all training data, and performance is obtained on the held-out test fold. Results reported in this paper are the average of 10 repetitions of this 10-fold nested CV process[5]. At each repetition, the data is shuffled, and a new fold partitioning is performed.

Five parameters are optimized in the inner loop of the nested CV procedure. In the random subspace selection procedure, there is the number of dimensions of each subspace ($|\mathcal{P}|$), the number of clusters (K) and the number of subspaces (R) generated. When creating the ensemble, the temperature (T) and the number of classifiers (M) in the ensemble also have to be selected. The robustness of the proposed system to these 5 parameters is studied in Section 2.7. The recommended parameter values of Section 2.7 are applied in experiments on the Newsgroups data sets. Thus, only two parameters were optimized.

The RSIS procedure does not depend on a particular clustering algorithm or base classifier. In this paper, SVM classifiers are used because of their good performance and versatility when used with kernels. The k-means algorithm is used for clustering because of its low computational complexity. The LIBSVM (Chang & Lin, 2011) library was used for the SVM implementation. A set of optimal parameters for the SVM classifiers was determined in a prior experiment by coarse grid-search via cross-validation on each data set. The exponential kernel was used in all experiments. For the synthetic data set, $C = 10$ and $\gamma = 10^{-1}$. For the Musk data sets, $C = 10$ and $\gamma = 10^{-6}$. The same settings were used for the Elephant and Tiger data sets, except with $\gamma = 10^{-3}$. For the Fox data set, $C = 100$ and $\gamma = 10^{-2}$ were used.

[5] Ten repetitions of a 10 folds CV is the protocol used in the vast majority of MIL publications.

Classification performance was compared using two metrics: the prediction accuracy, used in most papers in the literature, and the area under the ROC curve (AUC). Some authors advocate the use of the AUC over accuracy as a comparison metric for classifiers (Provost *et al.*, 1998; Ling *et al.*, 2003; Tax & Duin, 2008). When available, both are reported. To measure accuracy, a threshold β has to be optimized to maximize bag prediction accuracy once the pool of classifiers is created. Ideally, when enough data is available this is done on a held-out validation set. However, since the number of bags is limited in the benchmark data sets and our experiments showed held-out validation degrades performance. Therefore, the value of the decision threshold β was optimized on the training data. AUC is a global measure over all β values.

2.4.3 Reference Methods

Five well-known reference methods were implemented and tested for experiments with the synthetic data (see Section 2.5). These methods were selected because they yield good performances and represent a spectrum of different approaches that may perform differently depending on data set characteristics.

APR: This method was selected based on its popularity and its good performance on the Musk data sets. Zhou's MATLAB implementation (Zhou & Zhang, 2003) was used in the experiments. However, a modification was applied to obtain a classification score and compute the AUC. For each instance, the proportion of relevant dimensions in which the instance falls inside the hyper-rectangle is used as score. The score of a bag is given by the maximum instance score it contains. Preliminary experiments were conducted on data sets generated using the parameters listed in Table 2.3 with non-uniform negative distribution. The overall best results were obtained using $\tau = 0.99$ and $\varepsilon = 0.01$. These settings were used for all subsequent experiments on the synthetic data set. The recommended settings were used in the experiments on benchmark data sets.

Citation-kNN: This method was selected due to its popularity and good performance. Zhou's MATLAB implementation (Zhou & Zhang, 2003) was used, but the distance function was compiled to native code to decrease computation time. Also, to obtain a ROC curve, a score output, corresponding to the proportion of positive *citers* and *references*, was added to the function. Preliminary experiments were conducted on data sets generated using the parameters listed in Table 2.3 with non-uniform negative distribution. The overall best results were obtained using 5 citers and 5 references. These settings were used for all subsequent experiments on the synthetic data set. The recommended settings were used in the experiments on benchmark data sets.

mi-SVM: This method was selected because it is instance-based, uses SVM and is well-known. The LIBSVM (Chang & Lin, 2011) library was used for the SVM implementation. The decision values were used for AUC computation. The score of a bag is the highest decision value in the bag. An exponential kernel was used with parameters $\gamma = 0.1$ and $C = 10$. These settings were optimized via grid search in a preliminary experiment on data sets generated using the parameters listed in Table 2.3 with non-uniform negative distribution.

AL-SVM: This method was selected for comparison because it was showed to perform well on low witness rate problems. It is very similar to the mi-SVM algorithm because it minimizes the same objective function under the same constraints (Gehler & Chapelle, 2007). It is different in the way the algorithm is initialized and how labels are attributed by a deterministic annealing procedure, which is hoped to find a better solution. The authors provide an implementation of the algorithm which was used in the experiments. As suggested in the paper, the Gaussian kernel was used, and its width was set to the median pairwise distance between instances. The initial temperature was set to $10C$ and $C = 10$, as for mi-SVM.

CCE: The constructive clustering ensemble method (CCE) (Zhou & Zhang, 2007) was selected for comparison with the proposed method because both methods perform a clustering of the feature space and use an ensemble of SVM. At first, the feature space is clustered using a fixed number of clusters. Every bag is then represented by a binary vector, with each bit corresponding to a cluster. When at least one instance from a bag is attributed to a cluster, its corresponding bit is set to 1. The binary codes of the bags are used as feature vectors to train a classifier. Diversity is created in the ensemble by using a different number of clusters each time. The authors implementation is used in the experiment. This implementation uses *k*-means clustering and SVM classifiers. As recommended in the paper, the ensemble contains 5 classifiers and using 10, 20, 30, 40 and 50 clusters.

Reference Methods for Benchmark Data Sets: For experiments on benchmarking data (see Section 2.6), many reference MIL techniques are compared. In order to assess the benefits of the random subspace instance selection procedure, tests were also conducted using SVM ensembles in which the training subsets were composed of randomly selected instances. The algorithm is the same as the one proposed in Section 2.3, except that samples were drawn from bags with uniform probabilities. The results for MILES on the Newsgroups data sets were obtained using the MIL toolbox implementation (Tax & Cheplygina, 2015). The optimal hyper-parameters for MILES and mi-SVM were obtained via grid search using an inner loop cross-validation as described in Section 2.4.2.

2.5 Results on Synthetic Data

Experiments in this section show the robustness of the proposed RSIS method to various data set characteristics.

2.5.1 Number of Concepts

Figure 2.5 presents the performance of the proposed and reference methods with the synthetic data set when the number of concepts increases in the data set. As explained in Section 2.4.1, here, a concept refers to a data cluster or a distribution mode that may or may not be defined over the complete feature space.

The figure shows that the performance of APR is affected by the number of concepts in the data set. When there are many concepts, the algorithm either leaves some concepts outside of the hyper-rectangle, or encompasses all of them at the price of a greater false alarm rate. Moreover, this algorithm's performance depends on the geometry of distributions. While APR performs well with uniform negative distribution, it is not the case when the data is clustered. This can also be observed in Figures 2.6 and 2.7. When positive and negative distributions are multi-modal, the algorithm pursues two, sometimes, conflicting objectives. It must maximize the number of positive clusters contained in the hyper-rectangle, while minimizing the inclusion of negative ones. The spatial arrangement of these clusters varies with each generation of the data set, resulting in an higher deviation than with other algorithms in all experiments.

The mi-SVM algorithm is not vulnerable to multi-modal distributions. The use of a kernel enables the SVM to create disjoint data partitions without problems. mi-SVM performs better on multi-modal negative distributions, as opposed to APR and Citation-kNN because the structure of the negative data is informative in the instance label assignation process. This structure is however nonexistent when the negative distribution is uniform, which makes it more difficult to identify negative instances from other known negative instances. By comparing accuracy and AUC results in Figures 2.5, 2.6 and 2.7, one can see that the accuracy of mi-SVM can improve in many situations by optimizing the offset of the decision hyper-plane on bags instead of instances. For instance, with the uniform negative distribution, the accuracy is often about 50%, while the AUC results are competitive.

The AL-SVM algorithm is closely related to mi-SVM, as explained earlier. The AL-SVM has inherited some robustness to multi-modal distributions, however the deterministic annealing

procedure seems to make the algorithm overlook some concepts in the data when their number increases. This could be because the two algorithms are not initialized in the same way. In mi-SVM, all instances in positive bags are initialized as positive, while it is not the case for AL-SVM. If a majority of positive instances from the same concept are wrongly initialized as negative instances, the concept is never learned as positive. However, the deterministic annealing procedure has proved beneficial to find an the SVM hyper-plane offset in the case of the uniform negative instance distribution, where mi-SVM failed completely (see Figure 2.5 (a)). The performances of AL-SVM seem to always be inferior when considering the AUC. Inferior performances have also been observed by the authors when comparing the two algorithms on real-life benchmark data sets (Gehler & Chapelle, 2007).

In Citation-kNN, instances are assigned the same label as their bags, thus only negative instances may be mislabeled. When the negative distribution is uniform, the mislabeled negative instances are sparsely distributed across the feature space. Therefore, it is more unlikely that a majority of instances in a neighborhood will be mislabeled. However, when the negative instances are grouped in clusters, this particular situation becomes more probable. This explains the difference in the algorithm performance on the two versions of the data set. Citation-kNN appears to be somewhat resistant to the number of clusters in the non-uniform data set. However, a decrease in performance is observed in the uniform distribution case, but this may be due to the limited number of bags in the data set.

The ensembles created with the CCE procedure are affected by the number of concepts, but only in certain cases. If the positive and negative distributions are composed of clear clusters, the algorithm performs better than all others and obtains consistently near perfect results. A degradation is observed after 7 concepts, but an optimization of the number of clusters used in the clustering phase and the number of classifiers in the ensemble would probably perform better in these cases. However, CCE does not perform as well in situations where the negative distribution is not organized in clusters. This makes sense since the clustering, which is used to create the bag representation, has no clusters to find, and thus fails to create meaningful

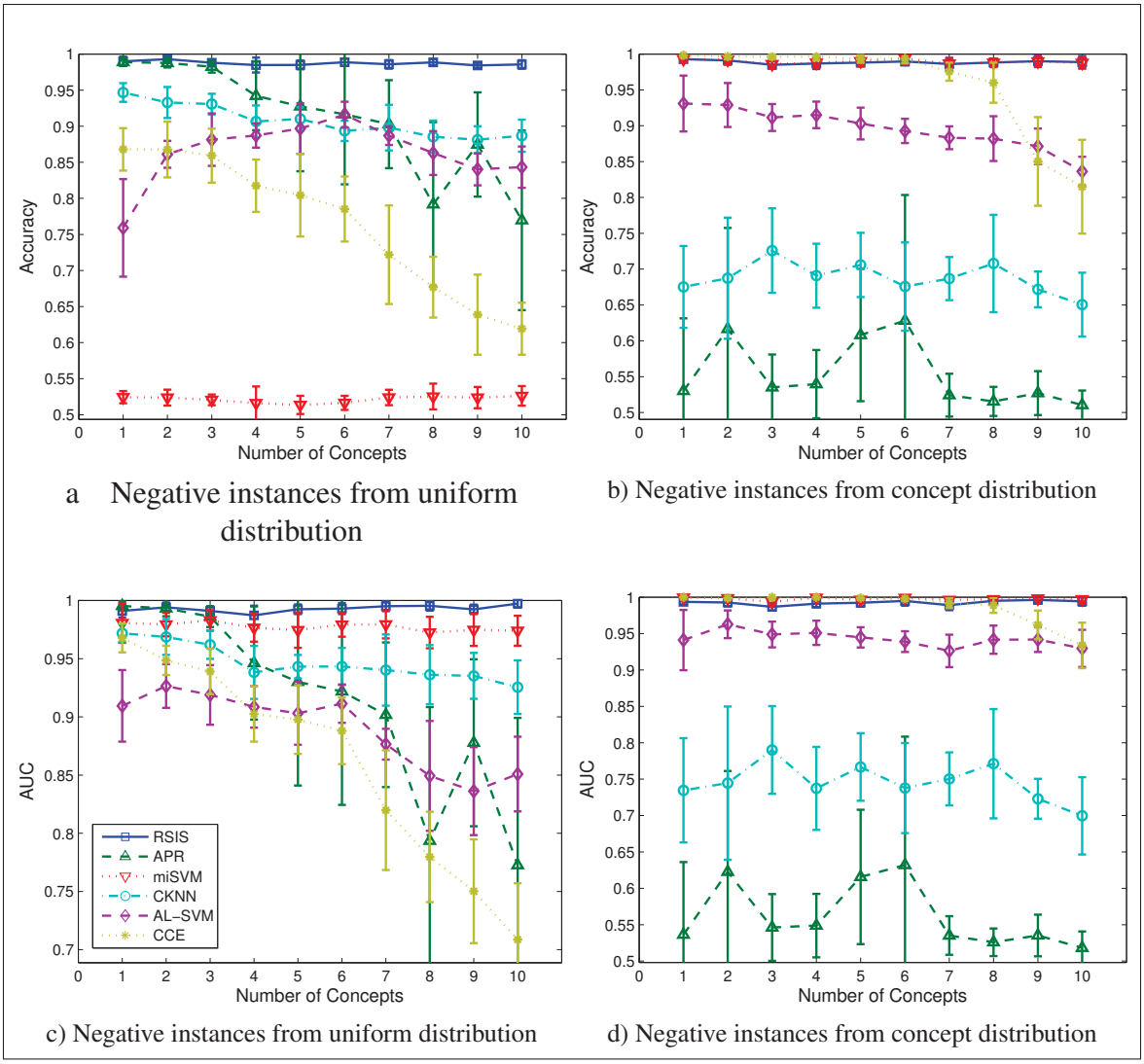


Figure 2.5 Average performance of EoSVM with RSIS and the reference methods for a growing number of concepts in the data set. The error bars correspond to the standard deviation. Results obtained in data sets where the negative distribution is uniform are in a) and c), while in b) and d), the negative distributions were composed of Gaussian clusters

feature vectors. Figure 2.5 (a) and (c) show that the problem worsen as the number of positive concepts increases.

Ensembles with RSIS are resistant to the number of concepts and outperforms reference methods. This is because, in the ensemble, the SVMs are not trained using the same positive data. All positive instances receive similar positivity scores, and thus have similar probabilities of

being selected as training instances. The fact that 5 clusters were used in the clustering process does not limit performance event if there are more cluster in the data set. Also, the shape of the negative distribution does not decrease performance as with the other methods. Comparable results were obtained using ensembles with randomly selected instances on this section. Since these ensembles already obtained near perfect results on this synthetic data, there was no room for significant improvement using RSIS. The efficiency of RSIS over random instance selection will be demonstrated on more difficult data sets in Section 2.6.

2.5.2 Witness Rate

Figure 2.6 presents results of obtained on the synthetic data when the witness rate is gradually increased. Some methods rely on the assumption that there is a majority of positive instances in positive bags. These methods thus perform well on certain types of data sets, such as the Musk data sets. This is the case for mi-SVM, because in the initialization, all instances in positive bags are assumed to be positive. However, as the proportion of positives declines, the more challenges arise to correctly identify the proper instance labels during the optimization process.

APR is also affected by the witness rate. The accuracy and AUC both increase as the witness rate rises. As with mi-SVM, instances from positive bags are considered positive. When the witness rate is low, there are more mislabeled instances, which leads to performance degradation. In the case of non-uniform distributions, the learning process does not converge with a very low witness rate. Deterministic annealing in AL-SVM provides a solution to these problems and thus, at lower witness rates, the AL-SVM performs better than mi-SVM.

Citation-kNN is also sensitive to the witness rate because, as stated earlier, instance labels correspond to bag labels. Hence, when the witness rate is low, there is a greater chance that a negative instance from a positive bag will cause a classification error. As for APR, performance rises almost linearly with the witness rate on the non-uniform negative distribution.

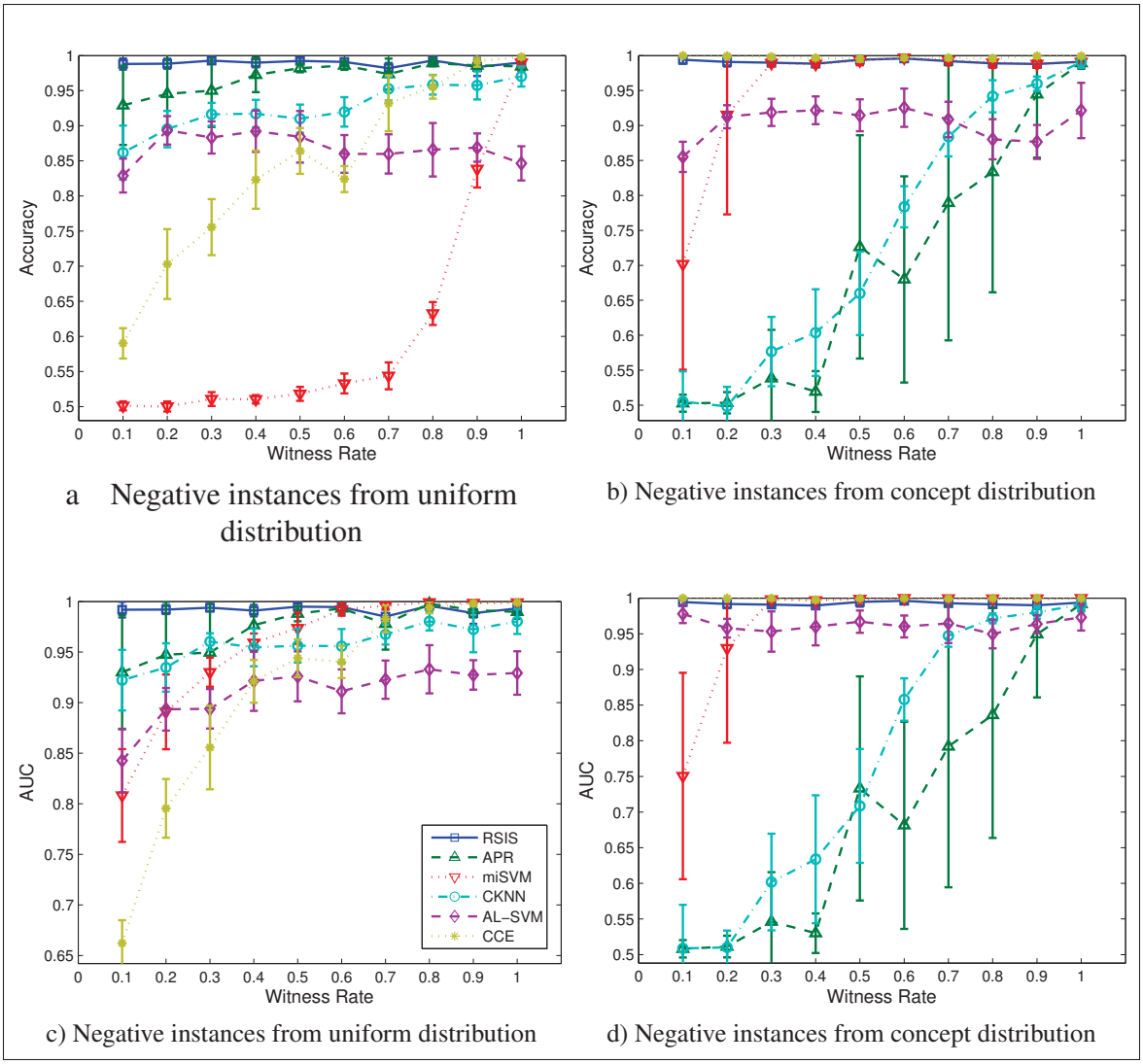


Figure 2.6 Average performance of ensembles with RSIS and the reference methods when varying the witness rate in the data set. The error bars correspond to the standard deviation. Results obtained in data sets where the negative distribution is uniform are in a) and c), while in b) and d), the negative distributions were composed of 3 clusters

As observed in the previous experiment, CCE has difficulties dealing with uniform distributions, however, when both distribution are composed of clear clusters, the algorithm works perfectly regardless of the witness rate.

Ensembles with RSIS performs consistently well under a wide range of witness rates. It is only outperformed by CCE with negative concept distributions and by mi-SVM when the witness

rate is very high. This Because RSIS selects the most probably positive instances for training. Only one instance per bag is selected. When all instances of positive bags are positive, the most difficult instances do not get picked as training instances. On the other hand, mi-SVM includes them in its model, and thus can achieve better performance in these particular cases. Also mi-SVM has lower computational complexity because only one classifier is used instead of an ensemble.

2.5.3 Proportion of Irrelevant Features

In Figure 2.7, the proportion of irrelevant features, was gradually increased to assess robustness to noise. An irrelevant feature is a feature which does not contain any information for a given concept. In other words, it is a feature in which instances, generated by given concept, are uniformly distributed. Irrelevant features are not the same for each concept, as illustrated in Figure 2.4, so feature extraction and selection techniques would not alleviate this challenge.

The performance of all of the tested methods decreased as the number of irrelevant features increased. In the non-uniform case, the accuracy of mi-SVM is rapidly affected by the inclusion of irrelevant features. However, the AUC results are as stable as the best performing algorithm, ensembles with RSIS. AL-SVM is affected in the same way as mi-SVM when considering AUC, but, as observed in previous experiment, the algorithm is better at determining the SVM hyper-plan offset. This can be observed through the higher accuracy of AL-SVM vs. mi-SVM in Figure 2.7(a). It also explains why the accuracy of AL-SVM degrades progressively in Figure 2.7(b), as opposed to the accuracy of mi-SVM.

Performance of Citation-kNN declines when a majority of features are uniformly generated. This algorithm depends on the Hausdroff distance, and when many irrelevant dimensions are considered in the distance calculation, the measure loses discrimination.

APR performance is affected by irrelevant features. In particular cases, the inclusion of irrelevant dimensions is beneficial (see Figure 2.7 (b) and (d)). When there are fewer relevant features, the probability that positive concepts share these features decreases. It is therefore

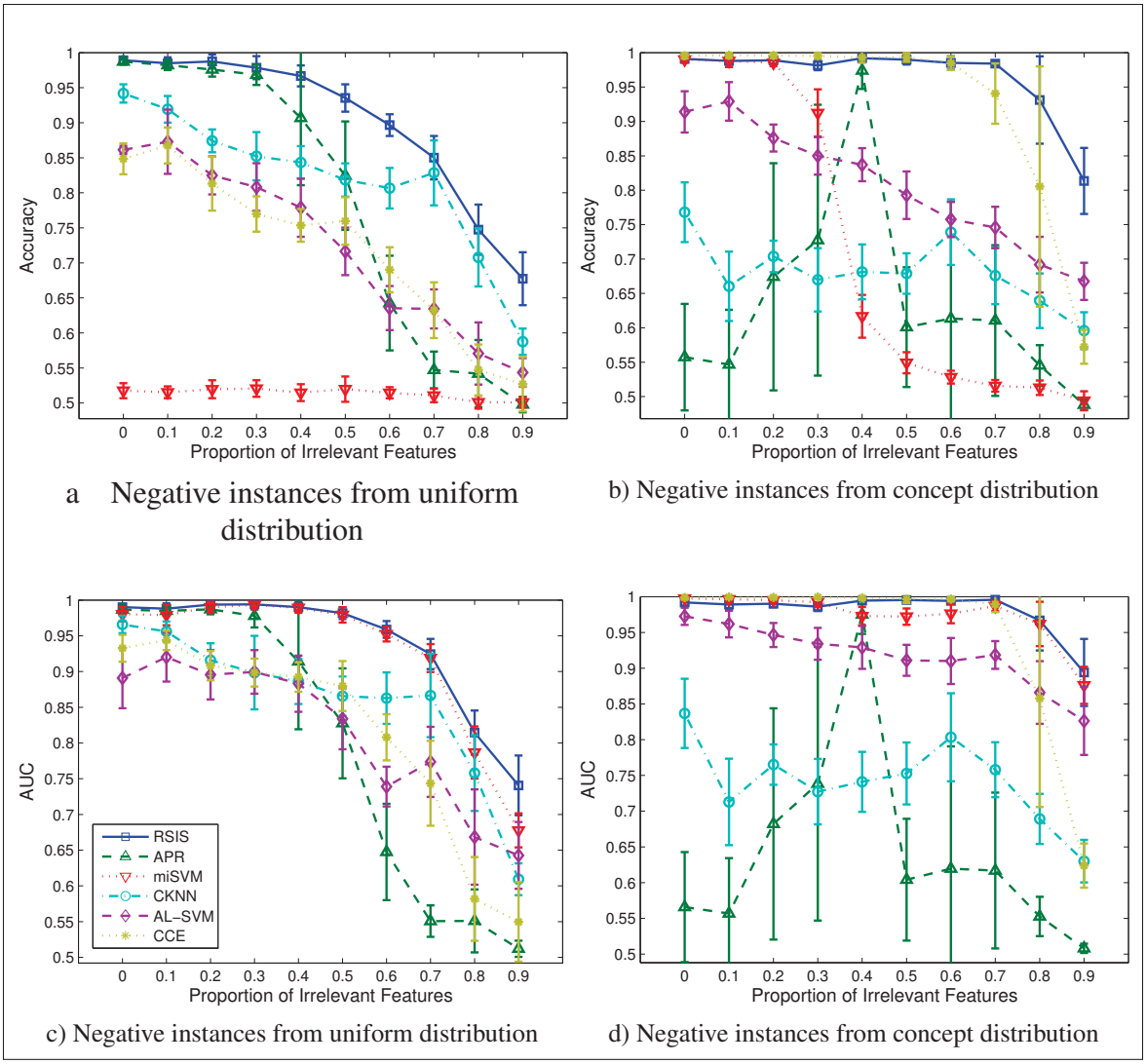


Figure 2.7 Average performance of ensembles with RSIS and the reference methods when varying the proportion of irrelevant features used to describe each concept. The error bars correspond to the standard deviation. Results obtained in data sets where the negative distribution is uniform are in a) and c), while in b) and d), the negative distributions were composed of 3 clusters

easier to define an hyper-rectangle that closely fits the positive instance distribution even if it is separated in distinct concept. This beneficial effect is however offset by ambiguity introduced by a high number of irrelevant features.

In this experiment, an irrelevant feature means a uniform distribution. This was showed to be the weakness of CCE, and this is why, even with the clustered negative distribution, performance drops drastically after the proportion of irrelevant features reaches 0.6.

As with the other algorithms, ensembles with RSIS are affected by irrelevant features. In this case, clustering is performed using the Euclidean distance. When a large proportion of feature in the data set is meaningless, the distance measure also becomes meaningless. However, by isolating features in subsets, the creation of random subspaces provides a certain resilience against this corrupting effect.

2.6 Results on Benchmark Data Sets

Experiments in the last section demonstrated RSIS robustness to different data set parameters. In this section, RSIS is compared to other methods on widely-used standard benchmark data sets. Results obtained with an ensemble of SVM (EoSVM) without the RSIS procedure are also presented to further assess the benefits of using RSIS.

2.6.1 Musk Data Sets

Results for RSIS on Musk data sets are reported alongside results from other alternatives in Table 2.4. Most papers do not provide the AUC, however results for a number of methods have been published in (Bunescu & Mooney, 2007b; Cheplygina *et al.*, 2015c; Ray & Craven, 2005; Cheplygina *et al.*, 2015a) and are reported here. Standard deviation is provided when available.

The results for Citation-kNN are obtained from Zhou's implementation (Zhou & Zhang, 2003), using the parameter settings suggested in the original paper ($C = 4$ and $R = 2$). New experiments had to be performed, because the original paper used leave-one-out cross-validation. This is also the case for CCE (Zhou & Zhang, 2007). The implementation provided by the authors was used, as well as the suggested parameters in the paper. The AUC for APR was also computed with Zhou's implementation, using the original paper's optimal parameters

($\tau = 0.999$ and $\varepsilon = 0.01$). The results for EM-DD come from (Andrews *et al.*, 2002), because the original paper optimized its results on the test data.

In light of the results reported in Table 2.4, one can see that RSIS delivers a similar or better accuracy on Musk1 than most other methods. Only APR and PC-SVM possess a statistically-significant advantage over RSIS. On Musk2, RSIS also delivers state-of-the-art results. Without standard deviations, it is difficult to assess the significance of the advantage of some methods. Nonetheless, MILIS outperforms RSIS with reasonable certainty (95%) on this data set.

When comparing based on the AUC, RSIS results are significantly superior to methods reported on Musk1. On Musk2, RSIS, MInD and MILES provide similar results, while outperforming the other techniques.

Finally, the results obtained with the SVM ensemble without the instance selection procedure are compared against ensembles designed with RSIS. The selection procedure significantly improve the accuracy performance of the ensembles. However, when comparing AUC, the results only differ on Musk1, suggesting that, without the selection procedure, the optimal classification threshold (β) is harder to determine. This is because, without instance selection, many classifiers in the ensemble are unreliable. While an optimal threshold works well with a certain data subset, varying performances will be obtained on different data.

2.6.2 Elephant, Fox and Tiger Data Sets

As for the Musk data sets, the accuracy of the original papers is reported along with the AUC, when available (Table 2.5 and 2.6). The results for APR, Citation-kNN and CCE were obtained with Zhou's implementations (Zhou & Zhang, 2003, 2007). RSIS performs better or as well as most methods reported on the Elephant data set. Only PC-SVM and mi-Graph have a statistically-significant advantage over RSIS. On the Tiger and Fox data sets, the results obtained with RSIS are surpassed by 4 and 5 methods, respectively. When comparing based on AUC, RSIS's results are superior or equivalent to all other reported methods.

Table 2.4 Experimental results on the Musk data sets.
Results from Bunescu & Mooney (2007), Cheplygina et al. (2015), Ray & Craven (2005)

	Accuracy (%)		**AUC**	
Algorithms	Musk 1	Musk 2	Musk 1	Musk 2
MILES (Chen *et al.*, 2006)	86.3 (1.4)	87.7 (1.4)	93.2 (2.9)	97.1 (1.6)
MILIS (Fu *et al.*, 2011)	88.6 (2.9)	91.1 (1.7)	-	-
APR (Dietterich *et al.*, 1997)	92.4 (2.7)	89.2 (3.0)	91.8 (1.0)	88.4 (2.6)
Citation-kNN (Wang & Zucker, 2000)	90.3 (1.3)	83.7 (2.3)	93.5 (2.0)	88.0 (1.9)
DD (Maron & Lozano-Pérez, 1998)	88.9	82.5	89.5	90.3
DD-SVM (Chen & Wang, 2004)	85.8	91.3	-	-
EM-DD (Zhang & Goldman, 2001)	84.8	84.9	87.4 (2.1)	86.9 (2.1)
mi-SVM (Andrews *et al.*, 2002)	87.4	83.6	93.9 (1.6)	81.5 (2.1)
MI-SVM (Andrews *et al.*, 2002)	77.9	84.3	91.5 (3.7)	93.9 (2.8)
MI-NN (Ramon, Jan and De Raedt, 2000)	88.0	82.0	-	-
Multinst (Auer, 1997)	76.7 (3.1)	84.0 (2.6)	-	-
RELIC (Ruffo, 2000)	83.7	87.3	-	-
MICA (Mangasarian & Wild, 2008)	84.4	90.5	-	-
AW-SVM (Gehler & Chapelle, 2007)	85.7	83.8	-	-
ALP-SVM (Gehler & Chapelle, 2007)	86.3	86.2	-	-
SVR-SVM (Li & Sminchisescu, 2010)	87.9 (1.7)	85.4 (1.8)	-	-
γ-rule (Li *et al.*, 2013)	88.4 (1.1)	84.9 (2.2)	-	-
MILBoost (Viola *et al.*, 2006)	69.8 (5.4)	76.4 (3.5)	74.8 (6.7)	76.4 (3.5)
MInD (Cheplygina *et al.*, 2015c)	-	-	93.4 (1.2)	95.4 (1.4)
TLC (Weidmann *et al.*, 2003)	88.7 (1.6)	83.1 (3.2)	-	-
MIBoosting (Xu & Frank, 2004)	87.9 (2.0)	84.0 (1.3)	-	-
PC-SVM (Han *et al.*, 2010)	90.6 (2.7)	91.3 (3.2)	-	-
MI-Graph (Zhou *et al.*, 2009)	90.0 (3.8)	90.0 (2.7)	-	-
mi-Graph (Zhou *et al.*, 2009)	88.9 (3.3)	90.3 (2.6)	-	-
MI-Kernel (NSK) (Gärtner *et al.*, 2002)	88.0 (3.1)	89.3 (1.5)	85.6	90.8
sbMIL (Bunescu & Mooney, 2007b)	-	-	91.8	87.7
stMIL (Bunescu & Mooney, 2007b)	-	-	79.5	68.4
CCE (Zhou & Zhang, 2007)	81.3 (2.0)	71.7 (3.4)	88.6 (1.4)	79.4 (3.4)
Diss. Ens. (Cheplygina *et al.*, 2015a)	89.3 (3.4)	85.5 (4.7)	95.4 (2.4)	93.2 (3.2)
EoSVM (random selection)	82.8 (1.9)	83.6 (2.0)	94.4 (1.3)	94.4 (1.1)
EoSVM (RSIS)	88.8 (1.3)	89.5 (1.6)	96.5 (0.9)	95.2 (1.0)

As was the case with the musk databases, there is a clear advantage of using RSIS over SVM without selection when comparing accuracy. In light of the AUC, however, a significant advantage is observed only on the Tiger data set. As for the results on the Musk data sets, these results suggest that RSIS produces a more reliable ensemble, which eases the selection of the final classification threshold.

Table 2.5 Experimental accuracy results on the Tiger, Fox and Elephant data sets

Algorithms	**Accuracy (%)**		
	Elephant	Tiger	Fox
MILES (Chen *et al.*, 2006)	79.0 (2.3)	81.0 (3.4)	62.5 (4.2)
APR (Dietterich *et al.*, 1997)	75.1 (1.3)	55.8 (1.1)	53.2 (1.2)
Citation-kNN (Wang & Zucker, 2000)	82.6 (0.9)	78.8 (1.3)	58.2 (1.1)
EM-DD (Zhang & Goldman, 2001)	78.3	72.1	56.1
mi-SVM (Andrews *et al.*, 2002)	82.2	78.4	58.2
MI-SVM (Andrews *et al.*, 2002)	81.4	84.0	57.8
MICA (Mangasarian & Wild, 2008)	80.5 (8.5)	82.6 (7.9)	58.7 (11.3)
AW-SVM (Gehler & Chapelle, 2007)	82.0	83.0	63.5
ALP-SVM (Gehler & Chapelle, 2007)	83.5	86.0	66.0
SVR-SVM (Li & Sminchisescu, 2010)	85.3 (2.8)	79.8 (3.4)	63.0 (3.5)
γ-rule (Li *et al.*, 2013)	84.4 (0.9)	80.8 (1.2)	62.8 (0.9)
MILBoost (Viola *et al.*, 2006)	79.5 (2.8)	78.5 (2.8)	63.0 (2.6)
PC-SVM (Han *et al.*, 2010)	89.8 (1.2)	83.8 (1.3)	65.7 (1.4)
MI-Graph (Zhou *et al.*, 2009)	85.1 (2.8)	81.9 (1.5)	61.2 (1.7)
mi-Graph (Zhou *et al.*, 2009)	86.8 (0.7)	86.0 (1.0)	61.6 (2.8)
MI-Kernel (NSK) (Gärtner *et al.*, 2002)	84.3 (1.6)	84.2 (1.0)	60.3 (1.9)
CCE (Zhou & Zhang, 2007)	79.6 (2.3)	75.6 (1.7)	61.5 (2.4)
Diss. Ens. (Cheplygina *et al.*, 2015a)	84.5 (2.8)	81.0 (4.6)	64.5 (2.2)
EoSVM (random selection)	82.5 (1.2)	73.7 (1.5)	57.9 (2.0)
EoSVM (RSIS)	84.6 (0.8)	82.5 (1.3)	61.1 (1.8)

2.6.3 Newsgroups

The results reported in Table 2.7 are taken from (Li & Sminchisescu, 2010) and (Zhou *et al.*, 2009). Tests were also conducted on the data sets using CCE, mi-SVM and MILES. As mentioned earlier, methods pooling all instances together, like MI-Kernel (Gärtner *et al.*, 2002), do not perform well when the witness rate is low. This is also the case for embedding methods, like MILES (Chen *et al.*, 2006). By considering the bags as a whole, these methods fail when a majority of instances do not belong to the target class. Results obtained in this section support this conclusion. The accuracy obtained with MILES and MI-Kernel does not exceed 60%, and often revolves around 50% for all data sets, which is the proportion of negative bags in the data set. Results obtained with mi-SVM are better. This method considers instances individually which seems to pay off in these low witness rate problems. The mi-Graph method derives an instance affinity matrix for each bag. This matrix is used to re-weigh the influence of instances

Table 2.6 Experimental results on the Tiger, Fox and Elephant data sets. Results from Bunescu & Mooney (2007), Cheplygina et al. (2015), Ray & Craven (2005)

	AUC		
Algorithms	Elephant	Tiger	Fox
MILES (Chen *et al.*, 2006)	88.3 (1.1)	87.2 (1.7)	69.8 (1.7)
APR (Dietterich *et al.*, 1997)	77.8 (0.7)	55.0 (1.0)	54.1 (0.9)
Citation-kNN (Wang & Zucker, 2000)	89.6 (0.9)	85.5 (0.9)	63.5 (1.5)
DD (Maron & Lozano-Pérez, 1998)	90.7	84.1	63.1
EM-DD (Zhang & Goldman, 2001)	88.5	72.3	67.6
mi-SVM (Andrews *et al.*, 2002)	84.3 (13.2)	83.3 (2.1)	56.1 (7.5)
MI-SVM (Andrews *et al.*, 2002)	90.7 (2.1)	87.2 (3.5)	68.7 (2.6)
MILBoost (Viola *et al.*, 2006)	89.0 (5.2)	84.1 (5.1)	61.1 (7.6)
MInD (Cheplygina *et al.*, 2015c)	93.1 (0.8)	85.1 (1.7)	60.5 (1.9)
MI-Kernel (NSK) (Gärtner *et al.*, 2002)	82.9	79.1	64.0
sbMIL (Bunescu & Mooney, 2007b)	88.6	83.0	69.8
stMIL (Bunescu & Mooney, 2007b)	81.6	74.5	60.7
CCE (Zhou & Zhang, 2007)	87.8 (1.1)	81.6 (1.8)	64.9 (2.6)
Diss. Ens. (Cheplygina *et al.*, 2015a)	92.3 (2.7)	87.8 (4.2)	70.2 (1.8)
EoSVM (random selection)	92.4 (0.7)	84.5 (1.3)	67.3 (1.4)
EoSVM (RSIS)	90.8 (0.8)	88.8 (0.9)	68.2 (1.8)

belonging to the same concept. Thus instances belonging to an under-represented concept in the bags gain more influence during classification. Using this scheme, the results obtained are slightly better than the results obtained with mi-SVM. CCE represents bags as a whole, but the representation is not directly based on the instance feature vectors. The feature vectors representing the bags encode only the presence, and not the quantity, of instances in different clusters. This provides a robustness to low witness rate because because the representation remains the same, independently of the number of similar negative instances in the bag. This is why despite using a bag-level representation CCE obtains competitive results. SVR-SVM is a method designed specially to withstand various witness rates. Therefore, the method yields far better results than MILES, MI-Kernel, mi-SVM and mi-Graph. The proposed method (RSIS) gets the best results on 16 of the 20 data sets. On 11 of the 20 data sets, it has a statistically significant advantage over all other methods. These results further illustrate the robustness to low witness rate of the proposed method.

Table 2.7 Experimental results on the Newsgroups data sets

	Algorithm Accuracy (%)						
Data Set	MILES	MI-Kernel	mi-SVM	mi-Graph	CCE	SVR-SVM	EoSVM
alt.atheism	55.9 (2.6)	60.2 (3.9)	79.2 (4.0)	65.5 (4.0)	77.8 (2.3)	83.5 (1.7)	**86.0 (1.8)**
comp.graphics	52.1 (2.9)	47.0 (3.3)	74.0 (3.2)	77.8 (1.6)	66.6 (1.8)	**85.2 (1.5)**	80.4 (1.4)
comp.windows.misc	50.5 (3.8)	51.0 (5.2)	62.3 (2.1)	63.1 (1.5)	59.9 (3.5)	66.9 (2.6)	**70.3 (2.7)**
comp.pc.hardware	49.9 (2.4)	46.9 (3.6)	59.3 (3.5)	59.5 (2.7)	66.2 (5.6)	70.3 (2.8)	**74.9 (2.2)**
comp.mac.hardware	52.2 (2.2)	44.5 (3.2)	75.4 (2.4)	61.7 (4.8)	61.4 (3.0)	**78.0 (1.7)**	**79.4 (2.4)**
comp.window.x	56.1 (2.0)	50.8 (4.3)	58.7 (4.0)	69.8 (2.1)	72.8 (3.5)	**83.7 (2.0)**	81.8 (1.6)
misc.forsale	53.3 (3.5)	51.8 (2.5)	68.9 (2.8)	55.2 (2.7)	63.2 (3.0)	**72.3 (1.2)**	**73.0 (2.3)**
rec.autos	50.5 (2.5)	52.9 (3.3)	61.0 (3.2)	72.0 (3.7)	65.9 (2.6)	**78.1 (1.9)**	75.0 (2.3)
rec.motorcycles	60.0 (3.2)	50.6 (3.5)	53.9 (1.7)	64.0 (2.8)	78.6 (2.0)	75.6 (0.9)	**80.0 (1.8)**
rec.sport.baseball	52.8 (2.8)	51.7 (2.8)	53.8 (2.5)	64.7 (3.1)	74.2 (1.2)	76.7 (1.4)	**87.1 (2.2)**
rec.sport.hockey	51.8 (1.6)	51.3 (3.4)	59.8 (3.8)	85.0 (2.5)	75.8 (2.1)	**89.3 (1.6)**	**90.5 (1.5)**
sci.crypt	56.4 (2.5)	56.3 (3.6)	67.3 (2.2)	69.6 (2.1)	72.9 (1.8)	69.7 (2.5)	**76.7 (1.6)**
sci.electronics	50.3 (1.6)	50.6 (2.0)	82.8 (3.2)	87.1 (1.7)	62.4 (2.3)	91.5 (1.0)	**93.7 (0.5)**
sci.med	54.4 (3.2)	50.6 (1.9)	69.9 (3.5)	62.1 (3.9)	72.2 (1.9)	74.9 (1.9)	**82.8 (2.5)**
sci.space	54.0 (4.0)	54.7 (2.5)	52.3 (1.7)	75.7 (3.4)	75.0 (2.3)	**83.2 (2.0)**	**81.0 (2.7)**
soc.religion.christian	56.7 (3.0)	49.2 (3.4)	50.0 (0.0)	59.0 (4.7)	76.6 (2.1)	**83.2 (2.7)**	80.6 (2.0)
talk.politics.guns	53.0 (4.3)	47.7 (3.8)	67.1 (2.8)	58.5 (6.0)	**73.4 (2.9)**	**73.7 (2.6)**	**74.5 (2.5)**
talk.politics.mideast	55.5 (4.5)	55.9 (2.8)	78.1 (1.9)	73.6 (2.6)	79.2 (2.4)	80.5 (3.2)	**85.0 (1.1)**
talk.politics.misc	59.2 (2.5)	51.5 (3.7)	67.6 (2.6)	70.4 (3.6)	**74.0 (2.2)**	72.6 (1.4)	**74.3 (1.9)**
talk.religion.misc	53.2 (1.9)	55.4 (4.3)	41.0 (1.6)	63.3 (3.5)	70.9 (3.1)	71.9 (1.9)	**75.5 (1.6)**

In several additional papers, only the alt.atheism in newsgroup data set is used a benchmark. Results are reported in Table 2.8. Most of the methods reported in this table yielded state-of-the-art results on at least one of the other benchmark data set, however, in this case, because the witness rate is below 2%, the performances of these methods decrease significantly. It shows that special care needs to be taken when designing a MIL algorithm used in low witness rate contexts. This is why SVR-SVM and RSIS yield the best performances.

2.7 Results on Parameter Sensitivity

In this section, experiments are conducted on the benchmark data sets to evaluate the parameter sensitivity of RSIS. The objective is to identify which parameters need careful tuning, and which parameters have a negligible effect on performance. The basic settings listed in Table 2.9 are varied one by one to observe their effect on performance. These settings were optimized on the Musk1 data set and then tested, as is, on the other databases to evaluate the specificity of the optimization procedure.

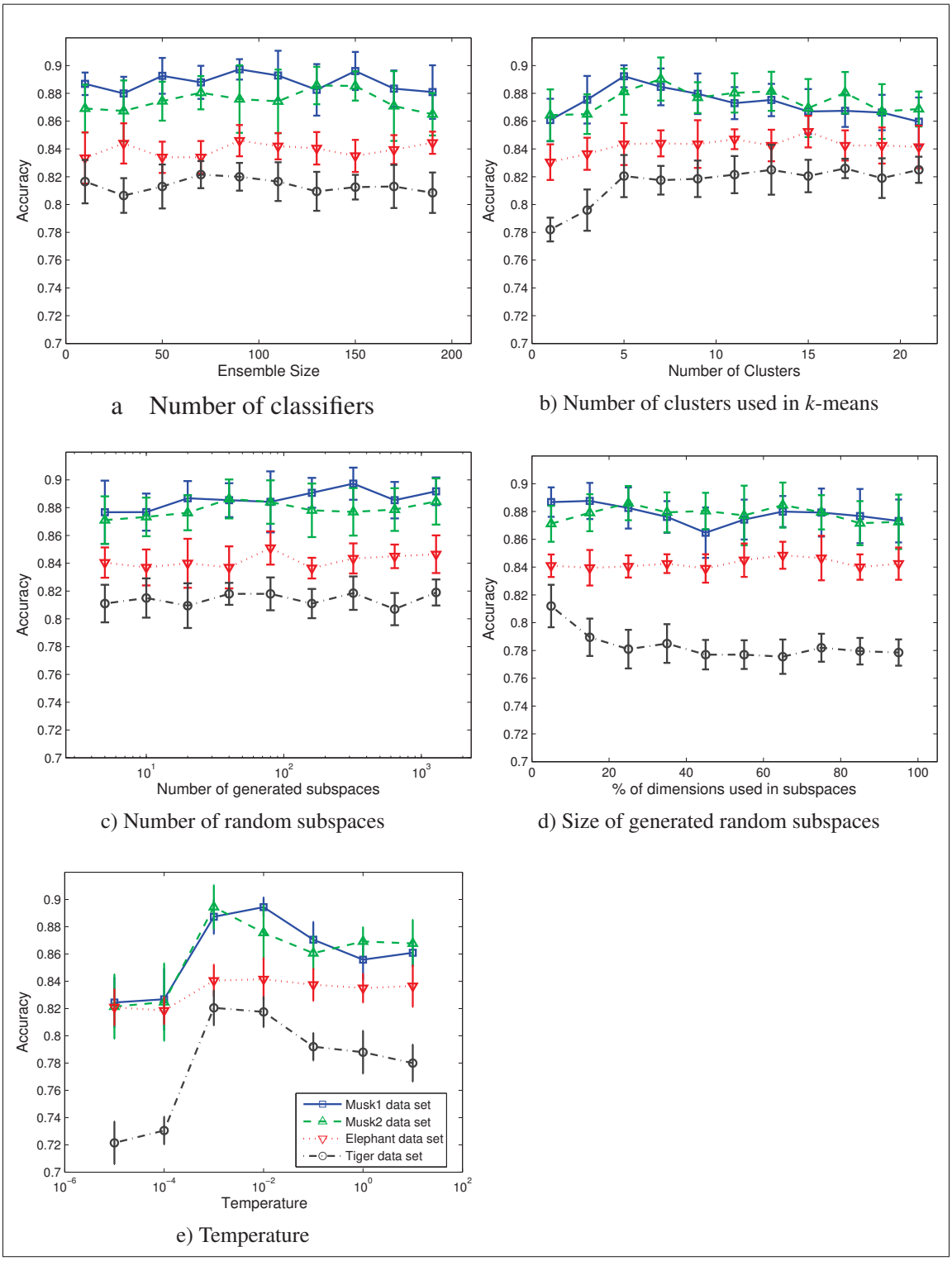


Figure 2.8 This figure presents a parameter sensitivity analysis of the proposed method on 4 benchmark data sets. In each graph, a parameter is varied and the average accuracy is reported. The error bars represent the standard deviation

Table 2.8 Experimental results on alt.atheism data set. Results from Zhou et al. (2009), Cheplygina et al. (2015), Li & Sminchisescu (2010)

Algorithms	**Accuracy (%)**
APR (Dietterich *et al.*, 1997)	49.0 (0.0)
Citation-kNN (Wang & Zucker, 2000)	50.0 (0.0)
Diss. Ens. (Cheplygina *et al.*, 2015a)	44.0 (4.5)
MI-SVM (Andrews *et al.*, 2002)	48.0 (2.0)
EM-DD (Zhang & Goldman, 2001)	49.0 (5.7)
MILES (Chen *et al.*, 2006)	55.9 (2.6)
MI-Kernel (Gärtner *et al.*, 2002)	60.2 (3.9)
mi-Graph (Zhou *et al.*, 2009)	65.5 (4.0)
Minimax-Kernel (Gärtner *et al.*, 2002)	76.0 (4.0)
CCE (Zhou & Zhang, 2007)	77.8 (2.3)
mi-SVM (Andrews *et al.*, 2002)	79.2 (4.0)
SVR-SVM (Li & Sminchisescu, 2010)	83.5 (1.7)
EoSVM (RSIS)	**86.0 (1.8)**

Table 2.9 Initial values in parameter sensitivity experiments

Parameter	**Symbol**	**Value**
Number of clusters	K	5
Temperature	T	0.01
Number of classifiers	M	100
Number of subspaces	R	500
Proportion of features used to create subspaces	$\lvert\mathcal{P}\rvert / \lvert\mathcal{F}\rvert$	5%

Figure 2.8 (a) shows that beyond 10, the number of classifiers M used in the ensemble does not significantly affect performance. For all data sets, the accuracy is contained in a maximum range of $\pm$ 1.0%. All of these values have a standard deviation between 0.7% and 2.5%. Moreover, except for some isolated cases (which do not represents a tendency), the accuracy falls in the standard deviation range of all other points on the curve. These small variations are mostly due to the randomness introduced by some parts of the algorithm and the cross-validation procedure.

Figure 2.8 (b) shows that the number of clusters K should be optimized based on the data. All curves indicate that a minimum of clusters should be used, but the optimal setting seems

to vary depending on the data set contents. Indeed, the quality of a clustering process using k-mean depends on the number of expected clusters (k) given the real number of clusters (Hamerly & Elkan, 2004).

It can be observed in Figure 2.8 (c) that the number (R) of subspaces generated does not offset performance significantly. Clearly a minimum number subspaces must be created otherwise performance degrades. However, this number is surprising low, as can be seen in Figure 2.8 (c). This suggests that the number of generated subspaces is not of paramount importance as long as a minimum number of 100 is met.

The number of dimensions per subspace $|\mathcal{P}|$ is defined in terms of proportion of the complete feature space. From Figure 2.8 (d), better results are obtained with less than 5% of the complete feature space, on the Tiger and Musk1 data set, while no noticeable difference can be seen on the other two data sets. Results indicates that smaller subspaces are generally preferred.

The temperature (T) is the most critical parameter, as seen in Figure 2.8 (e). When lower, the same instances are picked for each classifier of the ensemble, and the diversity is lowered, which degrades performance. On the other hand, if the temperature is higher, the selection process becomes more random, and incorrect instances are selected more often, which also degrades performance. This parameter should ideally be optimized for every problem.

Finally, it can be seen from Figure 2.8 that the results obtained on the other data sets using the Musk 1 configuration are comparable to those obtained in Section 2.6.1 and 2.6.2 with parameter full optimization. This supports the claim that the algorithm is insensitive to most parameter settings. As for T and K, the optimal settings for the Musk1 data set are a reasonable choice for the other data sets too. However, as shown in Figures 2.8 (b) and (e), marginally better accuracy may be achieved if these parameters are optimized. The recommendations of this section were successfully applied to the experiments on the Newsgroups data set (see Section 2.6.3).

2.8 Time Complexity

All experiments were conducted on a Intel i7/2.4 GHz processor with 8 GB of RAM. All algorithms have been implemented in MATLAB. However, the compiled implementation of LIBSVM was used for every SVM used in the experiments. Also, in CKNN, the computation of the Euclidean distance was compiled to native code.

The execution time were obtained on the Musk1 and Tiger data sets. Results reported in Table 2.10 are the average and the standard deviation of 10 repetitions of a 10-fold cross-validation. The training time does not include the time used for parameter selection since it is dependent on user-defined search grids. The number of parameters to be tuned is also reported for each method.

Ensembles with RSIS algorithms have more user defined parameters than all of the other methods because there are parameters to be set for the base learners (kernel type, γ and C) and for the ensemble. The user must set 5 parameters for this particular MIL implementation of ensembles with RSIS. Among the 5 RSIS parameters (see Table 2.9), only 2 are directly related to the new ensemble learning approach, and require careful tuning, while the others can be set as recommended (see Section 2.7).

In the computation of positivity scores, the time complexity for clustering random subspaces, when using the k-means algorithm, is given by $\mathcal{O}(ndKR)$ where K is the number of clusters, R is the number of random subspaces, and n and d are the number of instance and the data dimensionality, respectively. The training complexity of SVM is difficult to assess since it depends on the implementation and kernel. Using LIBSVM, it is empirically known that the computational complexity is higher than linear to the n (Chang & Lin, 2011). Here, it will be assumed to be $\mathcal{O}(n^2d)$. Since we train M classifiers, the complexity of the ensemble training phase is given by $\mathcal{O}(n^2dM)$. Along with the number of classifiers and the data dimensionality, the execution time depends on the regularization parameter C and the size of the data set (Shalev-Shwartz & Srebro, 2008). Testing time depends on the number of classifiers in

the ensemble (M), the data dimensionality and on the number of support vectors used in each SVM.

The training complexity of the mi-SVM is given by $\mathcal{O}(n^2dl)$, where l is the number of iterations needed by algorithm to converge. At each of these iterations, the SVM is retrained. The number of iterations required to obtain convergence is dependent on the nature of the data, and this is why the timing results exhibit high standard deviations. Compared to ensembles with RSIS, mi-SVM is faster to train with small data sets, but is slower as n increases. This is because the number of instances used to train each SVM of the ensemble is much smaller than the number used to train the one in mi-SVM. With RSIS, only one instance is selected in each bag to train the SVM, while every instances are used in mi-SVM. At some point, the complexity of mi-SVM ($\mathcal{O}(n^2dl)$) outgrows ensemble with RSIS complexity ($\mathcal{O}(B^2dM)$) where B is the number of bags. This can also be observed when comparing ensembles with RSIS and CKNN, which have a complexity of $\mathcal{O}(n^2dl)$. This suggests that ensembles with RSIS would scale better to big data sets. Independently of the data set size, during operation, ensembles with RSIS is the slowest of the four methods because every classifier in the ensemble needs to evaluate the instances in the bag. Finally, APR is the fastest method by far for training and testing regardless of the data set.

Table 2.10 Timing results on the Musk1 and the Tiger data sets

Data set	**Algorithms**	**Training time (ms)**	**Testing time (ms)**	**nb. of parameters**
Musk 1	APR	660 (65.0)	0.309 (0.672)	4
	CKNN	-	1290 (122)	2
	mi-SVM	62.9 (22.2)	3.82 (2.06)	3
	RSIS	963 (129)	935 (324)	2+3
Tiger	APR	137 (7.77)	0.541 (0.671)	4
	CKNN	-	22100 (381)	2
	mi-SVM	21200 (25700)	9.45 (2.16)	3
	RSIS	2040 (106)	4480 (373)	2+3

2.9 Conclusion

In this paper, a new instance selection mechanism using random subspaces is proposed to train MIL ensembles. The method can be used with any classifier and clustering algorithms. It is intended to be a versatile solution which can be applied to many types of MIL problems without extended knowledge on the data structure. This is because its performance is not affected by low and high witness rates, the shape of data distributions is of little impact on its performance, and it increases noise robustness. Moreover, the method is able to identify positive instances in bags which is sometimes required in MIL applications.

The proposed method was compared to state-of-the-art MIL methods on standard benchmark data sets, and yielded competitive results. A new synthetic data set was created to measure the adaptability of the proposed method to different data structures. The proposed method consistently yielded higher level of performance over the baseline methods for diverse conditions, namely witness rate, number of concepts and irrelevant feature rate. However, experiments suggest that other methods may perform better when the witness rate approaches 100%.

A drawback of the proposed method is the number of user-defined parameters to optimize. However, an analysis showed low sensitivity to most parameters. For instance, the number of generated subspaces is not critical, nor is the ensemble size. The number of dimensions used in subspaces should represent 5-10% of the complete feature space. This leaves only the temperature and the number of clusters in each subspace to be optimized. These recommendations were applied in the Newsgroups data set experiment and achieved state-of-the-art results. The recommendations were also applied to a synthetic data set, and consistently provided near-optimal results. As most ensemble methods, when compared with their single learner counterparts, the proposed method necessitates more processing time during operation. However, the proposed method has better training time scalability properties than mi-SVM and CKNN methods.

In future research, experiments should be conducted with different types of classifiers and clustering algorithms to measure the impact on performance. Also, in future versions of the algorithm, the number of instances selected in a positive bag could be adapted to the problem

characteristics. If an estimation of the witness rate can be obtained, selecting more than one instance per bag could increase performance of base-learners, and thus, increase ensemble performance. Also, a diversity measure applicable to MIL problems could enable the use of an ensemble selection mechanism used to prune redundant classifiers. Finally, experiments should be conducted to assess the suitability of RSIS as a preliminary instance labeling stage to increase robustness of existing algorithms. As stated before, many methods, such as mi-SVM, MIBoosting and MI-Kernel, initialize their optimization process assuming that all instances in positive bags are positive. Initializing these methods with RSIS could prove beneficial.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Quattriuum Inc.

CHAPTER 3

FEATURE LEARNING FROM SPECTROGRAMS FOR ASSESSMENT OF PERSONALITY TRAITS

Marc-André Carbonneau[1,2], Eric Granger[1], Yazid Attabi[1], Ghyslain Gagnon[2]

[1] Laboratory for Imagery, Vision and Artificial Intelligence,
École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] Communications and Microelectronic Integration Laboratory,
École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article under a second round of revision in IEEE Transactions on Affective Computing.
Initially submitted in August 2016.

Abstract

Several methods have recently been proposed to analyze speech and automatically infer the personality of the speaker. These methods often rely on prosodic and other hand crafted speech processing features extracted with off-the-shelf toolboxes. To achieve high accuracy, numerous features are typically extracted using complex and highly parameterized algorithms. In this paper, a new method based on feature learning and spectrogram analysis is proposed to simplify the feature extraction process while maintaining a high level of accuracy. The proposed method learns a dictionary of discriminant features from patches extracted in the spectrogram representations of training speech segments. Each speech segment is then encoded using the dictionary, and the resulting feature set is used to perform classification of personality traits. Experiments indicate that the proposed method achieves state-of-the-art results with an important reduction in complexity when compared to the most recent reference methods. The number of features, and difficulties linked to the feature extraction process are greatly reduced as only one type of descriptors is used, for which the 7 parameters can be tuned automatically. In contrast, the simplest reference method uses 4 types of descriptors to which 6 functionals are applied, resulting in over 20 parameters to be tuned.

3.1 Introduction

People spontaneously infer the personality of others from a wide range of cues. These cues may be visual, like facial expressions or posture, and may also be aural, like intonation patterns, choice of words or voice timbre. This assessment of personality traits naturally influences the way we interact with each other (Uleman *et al.*, 1996). The method proposed in this paper aims at performing this assessment automatically.

Being able to accurately predict the personality of an interlocutor is an important step toward better human-machine interactions. For example, people attribute personality traits to machines and interact differently with them depending on this perceived personality. For instance, extroverted people will interact longer with robots they perceive as extroverted (Tapus & Mataric, 2008). Detecting and understanding a person's personality would enable a machine to adapt its behavior to the user. It can also be used in e-learning applications by giving appreciative feedback on the personality projected by a user to improve its leadership or sale skills.

In the literature, five personality traits (the *Big-Five*) corresponding to psychological phenomenon are observable regardless of the situation and culture: openness, conscientiousness, extroversion, agreeableness and neuroticism (Digman, 1996). These traits influence the way people act and speak. For instance, in (Guadagno *et al.*, 2008) a correlation is established between openness and neuroticism and the probability of maintaining a blog. The choice of words by a subject based on his/her personality traits has also been studied in informal texts (Argamon *et al.*, 2005), conversations (Mairesse *et al.*, 2007) and on social media (Qiu *et al.*, 2012).

In the 2012 edition of the Interspeech competition on paralinguistics, one of the challenges was personality traits assessment from speech. This has motivated the proposition of several methods for this task. The baseline systems for the competition were designed using support vector machine (SVM) and random forest (RF) classifiers trained with 6125-dimensional feature vectors (Schuller *et al.*, 2012). They performed particularly well, and only two contestants were able to surpass their performance on the test set. It was observed that increasing the number

of features tends to increase recognition performance (Schuller *et al.*, 2012), thus large feature sets were extracted in the hope of capturing more of the relevant discriminant information. Some of the features were redundant or non-informative which motivated some contestants to use feature selection on the set of 6125 features (Chastagnol & Devillers, 2012; Wu, 2012; Pohjalainen *et al.*, 2012). The winners of the competition (Ivanov & Chen, 2012) added 21760 spectral features to the baseline feature set before performing selection.

Since 2012, the Interspeech competition 6125-dimension feature set of the baseline system has grown even larger. In 2015, it had increased to 6373-dimension (Schuller *et al.*, 2015a). Many of these features are statistics on the usual prosody features such as pitch, formants and energy, as well as more complex features, such as log harmonics to noise ratio, harmonicity and psycho-acoustic spectral sharpness. All of these application-specific feature extraction techniques require a fair knowledge and experience in speech processing to tune their parameters, select thresholds, pre-process data, etc. Moreover, results may vary from one implementation to another which limits the reproductibility of the experiments.

Many practitioners use software tools to extract prosody features, which accelerates the design of recognition solutions. However, even if these tools contain complete implementations of feature extraction algorithms, expertise in speech processing is required to configure the several parameters and options of each module. For instance, in openSMILE (Eyben *et al.*, 2013), one must choose between the cPitchACF (4 parameters) object and the cPitchShs object (9 parameters) to extract pitch, which in turn must be configured. The user may also use a pitch smoother, where four more parameters must be set. There are even more parameters to consider when extracting formants.

Aside from the complexity and variability of these feature extraction procedures, the use of large feature sets reduces the generalization capability of pattern recognition algorithms (Eyben *et al.*, 2016). Indeed, the exponential growth of the search space increases the amount of data needed to obtain a statistically significant representation of the data (Bishop, 2006). This represents a problem in affective computing application where data is limited because collec-

tion is costly. Moreover, smaller feature sets are desirable because they allow for faster training and classification.

The difficulties described above have been discussed by several researchers in the affective speech recognition community. The CEICES (Combining Efforts for Improving automatic Classification of Emotional user States) initiative attempted to create a standardized set of feature for emotion recognition in speech (Batliner *et al.*, 2006). The proposed set is a combination of 381 acoustic and lexical features selected from a pool of 4024 features that the authors have successfully used in their previous research. While the collection of features was standardized, the implementation of the feature extraction algorithms was not. Recently, another attempt has been made to reduce the size of the feature collection used for automatic voice analysis (Eyben *et al.*, 2016). A minimal number of descriptors were selected based on theoretical and empirical evidence. While the minimal and extended sets are compact (62 and 88 features respectively) several different algorithms are used for the extraction of the descriptors. These algorithms require expertise when tuning their various parameters[1].

In this paper, a method inspired by the recent developments in feature learning and image classification is proposed to alleviate these design choices for automatic assessment of personality traits. The temporal speech signals are translated into spectrogram images. Small sub-images, called patches, are densely extracted from these spectrogram images, and used during training to learn a feature dictionary yielding a sparse representation. The dictionary is used to encode each of the local patches. Each spectrogram is thus represented as a collection of encoded patches, which are pooled to create a histogram representation of the entire spectrogram. These histograms are used to train a classifier. During testing, a new speech signal is represented by a histogram, using the same dictionary, before classification.

The proposed method of representation, which is based on local patches, allows to capture paralinguistic information compactly. Because it encodes raw parts of the spectrogram images, the representation is richer than methods which characterize speech signals with statistics on the

[1] The feature set has been made publicly available through the openSMILE toolkit (Eyben *et al.*, 2013).

whole signal (Mohammadi & Vinciarelli, 2012; Eyben *et al.*, 2016; Schuller *et al.*, 2012). For instance, these methods use the mean, the standard deviation, kurtosis, min and max of the pitch or spectrum and cepstrum bins, which discard the relevant cues for personality assessment that the local shape of the signal contains. Moreover, when compared to these methods, the proposed method has fewer parameters, which can be more easily tuned using standard automatic hyper-parameter optimization techniques (e.g. cross-validation). In addition, the method inherits the robustness to deformation and noise of local image recognition methods applied to spectrogram analysis (Schutte, 2009; Sharan & Moir, 2015). Finally, since the dictionary learning process is performed in an unsupervised manner, additional training examples from other speech application domains can be used to learn a richer representation.

In essence, the proposed method leverages the power of representation inherent to sparse modeling, which learns features from the data. This approach generally leads to a high level of accuracy (Grosse *et al.*, 2007). The dimensionality of feature vectors needed for this level of performance is reduced by an order of magnitude when compared to the number of features used in the Interspeech challenges. Moreover, only one method is used for feature extraction which limits the number of parameters needing careful tuning. Finally, the proposed technique does not necessitate a feature selection stage which is usually time consuming during training.

The proposed method is compared to 6 reference methods on the SSPNet Speaker Personality Corpus used in the Interspeech 2012 competition. As stated in the overview of the challenge published in 2015 (Schuller *et al.*, 2015b), research in automated recognition of speaker traits is still active, and still requires much exploration to isolate suitable features and models for this task. In this regard, the novel technique proposed in this paper aims to provide a simpler alternative for extraction of a compact set of features that achieve state-of-the-art results.

The rest of the paper is organized as follows: The next section provides background information on feature learning in the context of speech analysis. Section 3.3 describes the proposed method. Section 3.4 presents the experimental data, protocol and reference methods. The results are analyzed in Section 3.5.1.

3.2 Feature Learning for Speech Analysis

Feature learning algorithms extract relevant features themselves, instead of relying on human-engineered representations, which are time consuming to obtain and are often sub-optimal. Feature learning has been used in several speech analysis applications. Some methods use deep neural networks, which intrinsically learn features, to perform automatic speech recognition (ASR) (Morgan, 2012; Mohamed *et al.*, 2012). These systems are not suitable for personality trait recognition because they analyze local time series (e.g. a phoneme), and fail to capture the global information in a speech segment. Deep learning has also been used for automatic emotion recognition. In (Trigeorgis *et al.*, 2016) a deep convolutional recurrent network learns a representation from the raw signal, while in (Kim *et al.*, 2013), the neural network learns a feature representation, not from the raw signal, but from a set of prosodic, spectral and video features. In (Deng *et al.*, 2013; Ghosh *et al.*, 2015), utterances where represented using sparse auto-encoders to perform emotion recognition. In (Heckmann *et al.*, 2011), base features were learned using independent component analysis on spectrograms. After a feature selection process, the selected features were combined in a higher hierarchical level, using non-negative sparse coding. These feature combinations were used with an hidden Markov model (HMM) to perform ASR.

Feature learning can be performed on several types of signal representation. When a speech signal is represented as a spectrogram, (i.e. concatenation in time of windowed Discrete Fourier Transform (DFT)), it can be analyzed through image processing. It has been demonstrated by neuroscientists that the same parts of the brain can be used to process both visual and audio signals (von Melchner *et al.*, 2000). This has motivated several researchers to investigate the application of image recognition techniques to spectrograms to analyze and recognize sound and speech signals. For example, histograms of oriented gradients (HOG) were used to perform word recognition (Muroi *et al.*, 2009). In (Dennis, 2014), spectrograms amplitudes are quantized and mapped into a color coded image. Color distributions are then characterized and analyzed. This method is inspired by content-based image retrieval methods (J.-L. Shih, 2002). In (Sharan & Moir, 2015), spectrograms and cochleograms are divided in frequency

sub-bands and analyzed as visual textures using gray-tone spatial dependence matrix features (Haralick *et al.*, 1973) alongside cepstral features. Audio spectrograms were employed with a convolutional deep Bayesian network, typically used for image recognition, to perform speaker identification and gender classification (Lee *et al.*, 2009) and with convolutional neural networks to perform emotion recognition on utterances (Mao *et al.*, 2014). The representation achieved a higher recognition performance when compared to mel-frequency cepstral coefficients (MFCC) and raw spectrograms. The Gabor function (sinusoidal tapered by a decaying exponential), were found to be good models of receptive fields in the human visual cortex (Marĉelja, 1980). This has motivated several authors to apply log-Gabor filter banks to spectrograms(Gu *et al.*, 2015; Buisman & Postma, 2012) to analyze paralinguistics.

A popular paradigm for image analysis is to extract features locally (instead of globally) from salient regions of an image, called patches. The set of patches, is used to represent an entire image. This type of approach, often called bag-of-words, have been successfully applied in numerous contexts for recognition in image (Philbin *et al.*, 2007; Csurka *et al.*, 2004) and video (Laptev *et al.*, 2008; Carbonneau *et al.*, 2015). Using local features in image recognition may lead to an increased robustness to intra-class variation, deformation, view-point, illumination and occlusion (Zhang *et al.*, 2006). When working with spectrograms, it translates to an increased robustness to noise (Schutte, 2009; Dennis, 2014). In (Matsui *et al.*, 2011) the SIFT descriptor was used to detect and encode key-points in spectrogram images of musical pieces to perform genre classification. Schutte proposed a deformable part-based model of local spatio-temporal features in speech recognition (Schutte, 2009). The method allowed to improve recognition performance over the HMM baseline system especially in the presence of noise.

Local-based methods in image recognition often exploit a set of predefined basis for decomposition such as wavelets, wedgelets and bandlets (Mallat, 2008). However, it has been shown that learning the basis directly on the data leads to a higher level of accuracy in several applications such as signal reconstruction (Elad & Aharon, 2006) and image classification (Raina *et al.*, 2007) and reconstruction (Aharon *et al.*, 2006). Based on these results, several recently

proposed spectrogram analysis methods learn representation on training data in order to benefit from the improved performance. For instance, in (Lyon, 2010) the spectrograms are segmented at different scales, and each segment is encoded as the most resembling word in a dictionary learned using the *k*-means algorithm. In (Yu & Slotine, 2009) the spectrograms of musical instruments are interpreted as visual textures. Sounds are represented by a vector encoding the resemblance between the spectrogram and a randomly constituted dictionary.

In the aforementioned dictionary-based methods, local descriptors are associated with the most representative code-word in the dictionary. Some algorithms use sparse coding to perform this association and learn a representation (Elad & Aharon, 2006; Peyré, 2009). Sparse coding is a type of feature learning which expresses a signal using a small number of basis from a learned set, usually called dictionary. Experiments have shown that encoding audio and visual signals using a sparse decomposition can lead to a high level of accuracy for various tasks such as acoustic event detection (Cotton & Ellis, 2011), speaker, gender and phoneme recognition (Lee *et al.*, 2009). Also, it was shown that a learned sparse representation of audio signals is akin to the early mammalian auditory system (Smith & Lewicki, 2006). This is why several recent methods use sparse coding to learn the dictionary and encode signals.

In the context of personality assessment from speech, paralinguistic cues must be analyzed globally. A personality trait is something that endures throughout entire speech segments belonging to the same speaker. This is different from many other speech recognition problems, like emotion recognition, where the target events have a relatively short duration. Methods used in other speech analysis applications, such as ASR and emotion recognition, do not typically capture global information from long speech segments. In most existing methods for personality recognition, this is achieved using statistical operators on low-level features. Unfortunately, this results in a high dimensional representation, which is prone to the curse of dimensionality, and require fair signal processing expertise to extract the low-level features. The proposed method represents a complete speech segment as an image then uses image recognition techniques, and thus, can perform global analysis. Moreover, it uses a feature learning approach,

which reduces the burden associated with feature engineering and yields a compact representation, and leads to increased recognition performance.

3.3 Proposed Feature Learning Method

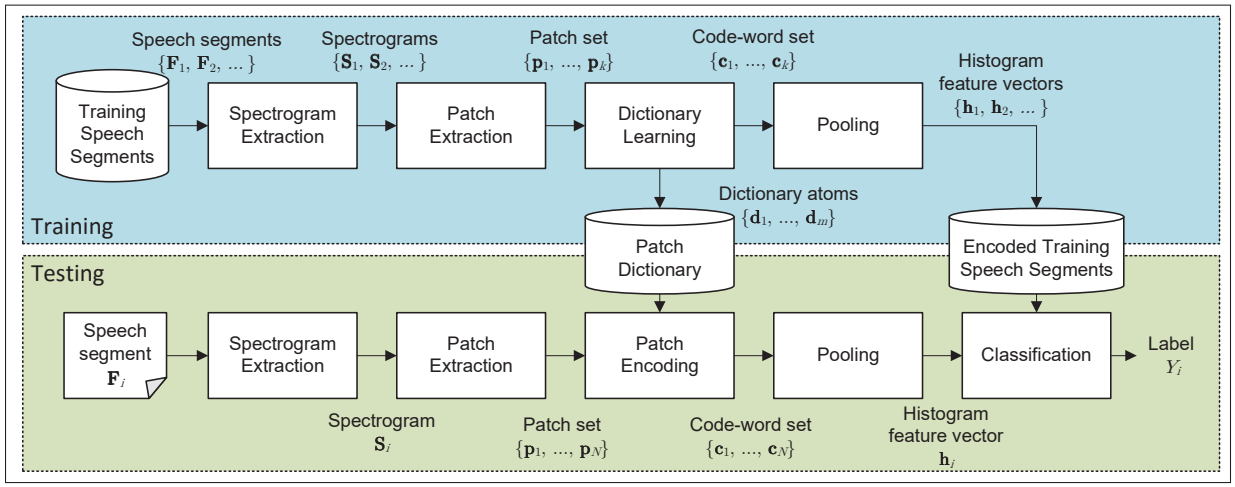


Figure 3.1 Block diagram of the proposed system for the prediction of a personality trait. The upper part illustrates the operations performed during training. The lower part illustrates sequence of operations performed to process an input speech sequence in test

This section presents a new method for predicting personality traits in speech based on spectrogram analysis and feature learning. The main stages of the proposed method are depicted in Figure 3.1. Specific details regarding our proposed solution for feature extraction, classification and dictionary learning are described in the next sections. The upper part is the pipeline for training. At first, for each speech segment $\mathbf{F}$ in the data set, a spectrogram $\mathbf{S}$ is extracted by applying a Fourier transform on a sliding window, yielding a 2-dimensional matrix. Small sub-matrices, called patches $\{\mathbf{p}_1,...,\mathbf{p}_k\}$ are then uniformly extracted from all the spectrogram matrices in the training set. A dictionary $\mathbf{D} = \{\mathbf{d}_1,...,\mathbf{d}_m\}$ is learned from these patches, and at the same time, the patches are encoded as sparse vectors called code-words $\{\mathbf{c}_1,...,\mathbf{c}_k\}$. A single m-dimensional feature vector representation $\mathbf{h}$ is obtained for each training speech sample by pooling together all code-words extracted from it. A two-class support vector machine (SVM) classifier is trained using these feature vectors for each personality trait.

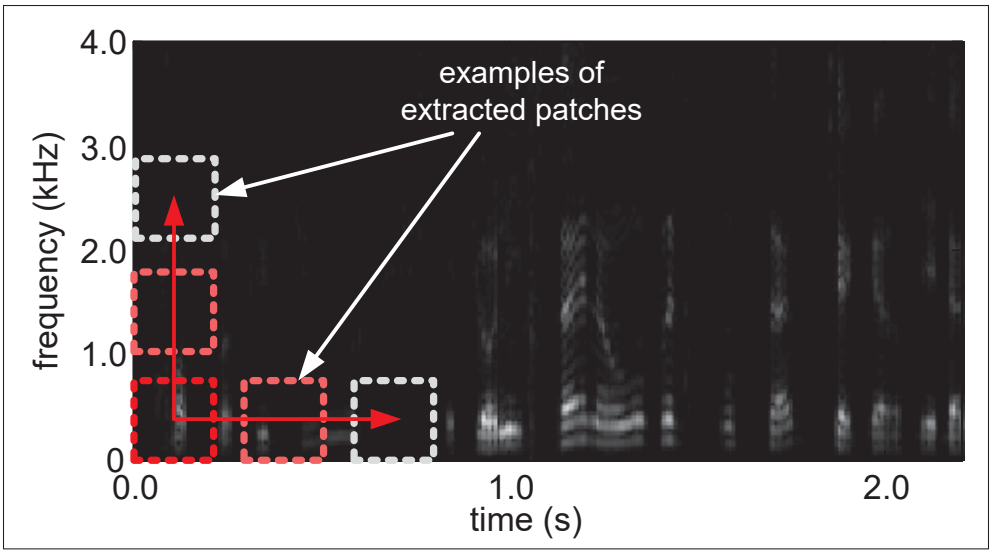


Figure 3.2 Example of spectrogram extracted from a speech file in the SSPNet corpus. White indicates high values while black indicates low values

The lower part is the pipeline used during testing, to predict a personality trait. Like in training, patches are extracted from the spectrograms. Each patch is encoded using the previously learned dictionary. The resulting code-words are then pooled to create a feature vector that is fed to a 2-class classifier to obtain a label Y representing to which end of the spectrum of a specific personality trait the speech segment corresponds.

3.3.1 Feature Extraction

Given a speech segment $x(n)$, the spectrogram $\mathbf{S}$ is the concatenation in time of its windowed DFT:

$$\mathbf{S} = \{\mathbf{X}_0, ..., \mathbf{X}_t, ..., \mathbf{X}_{T-1}\}, \tag{3.1}$$

where $\mathbf{X}_t$ is a column vector containing the absolute amplitude of the DFT frequency bins and T is the number of DFTs extracted from the signal. The absolute amplitude is favored over the log-amplitude as it has shown to yield better results for spectrogram image classification in (Dennis, 2014) and in our own experiments. The spectrograms are normalized: each frequency bin is divided by the maximum amplitude value contained in a time frame. This process results

in a 2-D matrix **S** which can be analyzed as a grey-scale image. An example of spectrogram extracted on the SSPNet Speaker Personality Corpus is illustrated in Figure 3.2.

From the matrix **S**, small patches, or sub-images, of $p \times p$ pixels are extracted at regular intervals. A vector representation $\mathbf{p}_i \in \mathbb{R}^{1\times d}$ of each patch ($d = p \times p$) is obtained by concatenating the value of all pixels. The vector $\mathbf{p}_i$ is encoded into $\mathbf{c}_i$ using a previously learned dictionary **D** containing m atoms (more details in Section 3.3.3). These atoms are vector basis that are used to reconstruct the patches. The code-vector $\mathbf{c}_i$ corresponding to the patch $\mathbf{p}_i$ is obtained by solving

$$l(\mathbf{c}_i) \triangleq \min_{\mathbf{c}_i \in \mathbb{R}^d} \frac{1}{2} \|\mathbf{p_i} - \mathbf{D}\mathbf{c}_i\|_2^2 + \lambda \|\mathbf{c}_i\|_1 \tag{3.2}$$

using the LARS-Lasso algorithm (Efron *et al.*, 2004). The loss function has two terms, each encoding an optimization objective, and λ is a parameter used to adjust the relative importance of the two terms. The first term is the quadratic reconstruction error, while in the second term, the ℓ_1 norm of the code vector is used to enforce sparseness. Once a code $\mathbf{c}_i$ is obtained for each patch $\mathbf{p}_i$, the absolute value of all the codes are summed to obtain a histogram **h** describing the entire spectrogram **S**:

$$\mathbf{h} = \sum_i |\mathbf{c}_i| \tag{3.3}$$

These histograms represent the distribution of patches over speech segments. It is thus possible to directly compare segments of different length.

3.3.2 Classification

The speech segments are represented by histograms and thus, appropriate distance measure should be employed. Several distance measures have been proposed to compare histograms. In this paper's implementation, the χ^2 distance is used because it showed competitive performance for visual bag-of-words histograms (Zhang *et al.*, 2006). The χ^2 distance is given by:

$$d(\mathbf{g}, \mathbf{h}) = \sum_{i=1}^{m} \frac{(g_i - h_i)^2}{g_i + h_i}, \tag{3.4}$$

where g_i and h_i are the i^{th} bins of histograms $\mathbf{h}$ and $\mathbf{y}$, and m corresponds to the number of words in the dictionary.

In this paper d is used in an SVM framework with an exponential kernel (Chapelle *et al.*, 1999):

$$k(\mathbf{g},\mathbf{h}) = e^{-\gamma d(\mathbf{g},\mathbf{h})}, \tag{3.5}$$

where the parameter γ controls the kernel size.

While the implementation of this paper employs the χ^2 distance and an SVM classifier, the proposed methods is not bound to these choices, and other distance functions and classifiers can be used.

3.3.3 Dictionary Learning

The objective of the dictionary learning phase is to generate a representative dictionary $\mathbf{D} = [\mathbf{d}_1,...,\mathbf{d}_m] \in \mathbb{R}^{d\times m}$ given the matrix $\mathbf{P} = [\mathbf{p}_1,...,\mathbf{p}_k] \in \mathbb{R}^{d\times k}$ containing patch vectors extracted from the training set. Generally, for image classification tasks, best results are obtained with over-complete ($m > d$) dictionaries (Tosic & Frossard, 2011).

A dictionary of atoms $\mathbf{D}$ and sparse code-words $\mathbf{C}$ can be obtained by minimizing the following loss function:

$$l(\mathbf{C},\mathbf{D}) \triangleq \min_{\mathbf{C}\in\mathbb{R}^{m\times k},\mathbf{D}\in\mathcal{C}} \frac{1}{2}\|\mathbf{P}-\mathbf{D}\mathbf{C}\|_2^2 + \lambda\|\mathbf{C}\|_1 \tag{3.6}$$

In this equation, λ is the same as in (3.2) and is used to adjust the weight of the sparseness term in the loss equation. The convex set:

$$\begin{aligned}\mathcal{C} \triangleq \{\mathbf{D}\in\mathbb{R}^{d\times m} \textbf{ s.t. } \forall i = 1,...,m, \mathbf{d}_i^T\mathbf{d}_i \leq 1 \\ \textbf{and } \forall i = 1,...,m, \mathbf{d}_i \in \mathbb{R}_{\geq 0}\}\end{aligned} \tag{3.7}$$

enforces two constraints. The first is used to restrict the magnitude of the dictionary atoms. The second is used to make sure each element of each atom in the dictionary is positive. Since

the spectrogram is purely positive, better results are obtained by enforcing this constraint. The joint optimization of **C** and **D** is not convex. However if one term is fixed the problem becomes convex. Thus, a common strategy is two alternate between updating **C** while **D** is fixed and updating **D** while **C** is fixed until a stopping criterion is met (Lee *et al.*, 2006).

Figure 3.3 shows an example of dictionary atoms learned using the above described procedure. Some atoms encode short intonation patterns with ascending and descending linear patterns, while others encode more punctual accents which may help discriminate personalities based on speech energy variation. The audio files from the SSPNet Speaker Personality Corpus were used to learn the atoms. The same dictionary can be used for all traits.

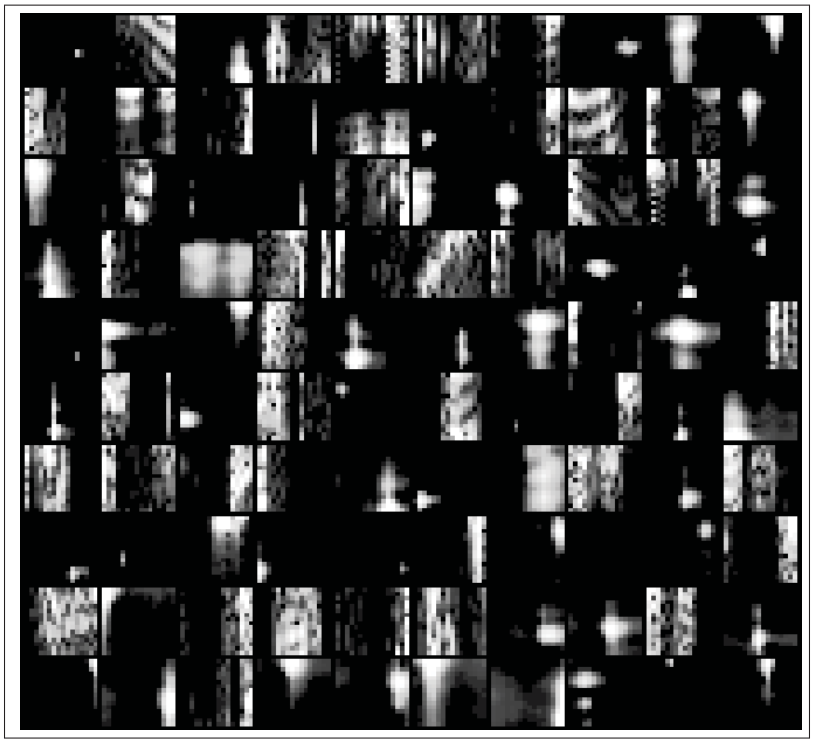


Figure 3.3 Example of patches from a dictionary created with sparse coding

3.4 Experimental Methodology

The SSPNet Speaker Personality corpus (Mohammadi & Vinciarelli, 2012) is the largest and most recent data set for personality trait assessment from speech. It consists of 640 audio clips randomly extracted from French news bulletins in Switzerland. All clips have been sampled at 8 kHz and most of the clips are 10 seconds long, but some are shorter. Each clip contains only one of the 322 different speakers. Eleven judges performed annotation on each clip by completing the BFI-10 personality assessment questionnaire (Rammstedt & John, 2007). From the questionnaire a score is computed for each of the *Big-Five* personality traits. Precautions were taken to avoid sequence and tiredness effects in the annotation process. The judges did not understand French and therefore were not influenced by linguistic cues. In (Mohammadi & Vinciarelli, 2012) the assessment of the judges were considered as positive if the score was greater than 0 and negative otherwise. The labeling scheme was refined for the competition (Schuller *et al.*, 2012). In this case, an assessment was considered positive if the score given by a judge was higher than the average score given by this particular judge for the trait. In both cases, the final label for an instance was obtained by a majority vote from all of the 11 judges. Preliminary experiments showed a 1$\sim$2% difference in accuracy performance between the two labeling schemes. The results reported in this paper were obtained using the competition's labeling scheme.

The metric used to compare accuracy is the unweighted average recall (UAR), which is the same as in the competition. The UAR is the mean of each class accuracy, and thus is unaffected by class imbalance. To assess performance, a 3-fold cross-validation procedure was used to limit the effect of sampling-induced variance in the results. Precautions were taken to make sure that all samples belonging to the same speaker are grouped in the same fold. Sampling-induced variance effects were observed in the Interspeech 2012 Speaker Trait challenge. The results obtained for the conscientiousness trait with the development partition are lower than the results obtained with the test partition. For instance, the baseline method using SVM obtained a UAR of 74.5% in training, but increased to 80.1% in testing (Schuller *et al.*, 2012). The same phenomenon was observed with the random forest classifier (74.9%

to 79.1%). This suggests that the test data may have been easier to classify than the average data. This hypothesis is supported by the fact that the results obtained using a cross-validation procedure in (Mohammadi & Vinciarelli, 2012) were also closer to 70% than 80%. Nested cross-validation (Stone, 1974) was used to optimize the hyper parameters for all classifiers and the dictionary learning parameters (dictionary size and λ). In nested cross-validation, an outer cross-validation loop (3 folds) is used to obtain the final test results, and an inner loop (5 folds) is used to find the best hyper parameter via grid search. Hyper-parameter optimization is thus performed for each of the 3 test folds separately.

For the proposed method, spectrograms were extracted using a short-time Fourier transform with a 128 sample Hamming window. This translates into 16 ms segments at the sample rate (8 kHz) of the Speaker Personality corpus. There was a 75% overlap between two successive speech segments. The extracted patches were 16×16 *pixels*, yielding 256-dimensional feature vectors. A new patch was extracted each 8 time steps and each 4 frequency bins. All of these 5 parameters (FFT window size and overlap, window type, patch size and stride) were selected based on preliminary experiments and were not subsequently optimized. Only 2 parameters, the dictionary size $\in \{100, 200, 400, 800\}$ and $\lambda \in \{0.05, 0.10, 0.20, 0.30, 0.40, 0.50\}$, were optimized in the experiments using the aforementioned cross-validation scheme. An importance weighting scheme was used to deal with class imbalance (Rosenberg, 2012). This was achieved by attributing different misclassification cost in the SVM hinge loss function to the target classes. The cost for the positive class was multiplied by a factor corresponding to the class imbalance ratio. The SPAMS toolbox (Mairal *et al.*, 2009) was used for dictionary learning and encoding and LIBSVM (Chang & Lin, 2011) was used for the SVM implementation.

Three reference methods were selected to compare performance. The methods were chosen because they are well documented and can be reproduced without ambiguity. The first method was proposed by Mohammadi & Vinciarelli in (Mohammadi & Vinciarelli, 2012). Prosody features were extracted using Praat (Boersma & Weenink, 2001), the same software used in the original paper. The low-level feature extracted were pitch, first two formants, energy of speech, and length of voiced and unvoiced segments. The features were extracted using 40 ms

long windows at 10 ms time steps. The features were whitened based on means and standard deviations estimated on the training folds. Four statistical properties were then estimated from the 6 prosody measures yielding a 24-dimensional feature vector for each speech file. The statistical features were the minimum, maximum, mean and the entropy of the differences between consecutive feature values. As in (Mohammadi & Vinciarelli, 2012), an SVM and a logistic regression (LR) were used for classification. The logistic regression implementation of the MATLAB Statistic and Machine Learning Toolbox was used. For the SVM, the LIBSVM implementation was used with the linear and the radial basis function (RBF) kernels.

The second method is the baseline used in the Interspeech 2012 speaker trait challenge (Schuller *et al.*, 2012). The 6125 low-level features were extracted using the openSMILE software (Eyben *et al.*, 2013) with the preset named after the challenge. The features were whitened based on means and standard deviations estimated on the training folds. For the linear SVM, the LIBSVM implementation (Chang & Lin, 2011) was used which performs sequential minimal optimization, the optimization algorithm used in the challenge baseline. The use of Gaussian kernel was also explored but did not yield better results. For the RF classifier, MATLAB implementation from the Statistic and Machine Learning Toolbox was used. This method was selected because it yield state-of-the-art performance. Only 2 of the methods proposed in the challenge outperformed the baseline with a UAR margin of 0.1% for (Montacié & Caraty, 2012) and of 1% for (Ivanov & Chen, 2012), which is not significant.

The third and most recent benchmark method uses the features prescribed in the Geneva minimalistic acoustic parameter set (GeMAPS) (Eyben *et al.*, 2016). The minimalistic set can be extended (eGeMAPS) by including MFCC coefficients, spectral flux and additional formant descriptors. The features were extracted using the preset supplied in openSMILE. Classification was achieved by a linear SVM using the LIBSVM implementation. The hyper-parameters were optimized in the same way as for the Interspeech method. This method was selected because it is intended to reduce the complexity of the feature extraction stage in paralinguistic problems, same as the proposed method.

Finally, we replaced the feature learning algorithm in the proposed method by sparse auto-encoders (SAE) and stacked sparse auto-encoders using an implementation similar to (Deng *et al.*, 2013). The topology and loss function parameters were optimized using random search as prescribed in (Bergstra & Bengio, 2012) because the number of hyper-parameters is too high to perform grid search in reasonable time. The number of neurons on each layer ranges from 50 to 800. A sample pool of 200k patches were used for training the SAE. Sparseness and regularization weights and parameters were sampled from log-uniform distributions.

3.5 Results

3.5.1 Accuracy

Table 3.1 Performance on the SSPNet Speaker Personality corpus. Legend: O = Openness, C = Conscientiousness, E = Extroversion, A = Agreeableness and N = Neuroticism

	Unweighted Average Recall (%)					
Algorithm	O	C	E	A	N	Avr.
Mohammadi & Vinciarelli (LR)	56.1	69.6	72.4	55.7	67.4	64.2
Mohammadi & Vinciarelli (SVM)	57.7	68.0	74.3	57.4	65.5	64.6
Interspeech Challenge Baseline (SVM)	**58.7**	69.2	74.5	62.2	69.0	66.7
Interspeech Challenge Baseline (RF)	52.9	69.0	**77.5**	60.1	68.2	65.5
GeMAPS (SVM)	56.3	72.2	74.9	61.9	68.9	66.8
eGeMAPS (SVM)	53.7	**72.5**	75.1	62.0	66.6	66.0
SAE 1-Layer (SVM)	57.1	64.3	69.2	62.0	65.8	63.7
SAE 2-Layers (SVM)	57.3	63.6	69.0	60.3	61.9	62.4
Proposed Method	56.3	68.3	75.2	**64.9**	**70.8**	**67.1**

The performance of the proposed and baseline methods on the SSPNet Speaker Personality corpus is reported in Table 3.1. The best average UAR was obtained using the proposed method. However, the results obtained when using the challenge features and GeMAPS with an SVM classifier are comparable. The method proposed by Mohammadi and Vinciarelli yields slightly lower accuracy than the other methods, although the difference in performance in most cases

is small and may be negligible. Particularities in the data set and the type of classifier, as well as its implementation, are most likely the reason for these variations in performance. For instance, using the same features and a different classifier, the Interspeech 2012 challenge baseline (Schuller *et al.*, 2012) obtains a UAR of 58.7% (SVM) and 52.9% (RF) for the openness trait.

The performance gap between the proposed method and SAE is due in part to the way sparsness is enforced in the optimization loss function. SAE use the Kullback–Leibler divergence (Deng *et al.*, 2013) of the neuron activation proportion and a fixed parameter, while the proposed method uses the ℓ_1 norm of the code vector. SAE represents complex intonation patterns with a combination of more generic patches while the proposed method tends to encode these complex patterns with single patches. The sum pooling process hides the discriminative information of intonation patterns represented as a composition of generic patches.

There are differences between the representations. For instance, the proposed method is not well adapted to represent pitch nor speech rate. Estimating the pitch is difficult because once the patches are extracted, their location is discarded. In contrast, all reference methods explicitly extract pitch and compute statistics on the measure. Speech rate is also difficult to represent by the proposed method since patches encode local information while speech rate is more of a global measure. All reference methods capture speech rate better because they extract statistics on the length and proportion of voiced and unvoiced segments. This slightly impedes the proposed method for the recognition of the openness trait, for which pitch and speech rate have been identified as markers (Mairesse *et al.*, 2007; Addington, 1968). It could explain the 2.4% and 1.4% difference between the proposed and reference methods using SVM. However, these two markers are also indicative of neuroticism (Mairesse *et al.*, 2007), and the proposed method performs well on this class. This could be explained by its ability to capture voice timbre and short intonation patterns. The proposed method uses raw chunks of the sound spectrogram as representation, and thus can capture this kind of information with high fidelity.

Table 3.2 Parameter complexity of the methods

Algorithm	Number of			
	Features	Descriptors	Functionals	Parameters
Mohammadi & Vinciarelli	24	4	6	>20
Interspeech Challenge Baseline	6125	21	39	>200
GeMAPS	62	13	10	>100
eGeMAPS	88	16	12	>100
SAE 1-Layer	100-800	1	1	>30
SAE 2-Layers	100-800	1	1	>30
Proposed Method	200-800	1	1	7

3.5.2 Complexity

While accuracy is generally similar for all methods, the main advantage of the proposed method is the important reduction of effort and design choices needed for its implementation. The amount of human expert intervention is different for all methods as reported on Table 3.2. In the proposed method, only 1 feature extraction algorithm was used instead of 4 for (Mohammadi & Vinciarelli, 2012), more than 10 for GeMAPS and over 20 in (Schuller *et al.*, 2012). In addition, in these reference methods, a set of functionals were applied to the extracted features. Some of these functionals were simple measures like mean, min/max and standard deviation, but others were more complex and parametrizable. For instance, functionals relying on peak distance need a peak detector that has to be fine-tuned. These feature extraction algorithms require parametrization which must be performed by a signal processing expert. A similar argument applies to SAE. These models necessitate a fair amount of expertise and experience to choose the appropriate topology and loss function, to tune the numerous hyper-parameters and to configure the optimization algorithm. Also, when compared to the baseline of the Interspeech challenge, the feature set used in the proposed method is much smaller (at most 800 features instead of 6125). Smaller feature sets are desirable because they reduce algorithmic complexity, and are less subject to problems associated with the curse of dimensionality.

During training, the time complexity of the proposed method is higher than for the other methods because of the dictionary learning phase. However, at test time, less operations are required

than for all other methods except SAE. In the proposed method, two main operations are carried out: spectrogram extraction and patch encoding. Spectrogram extraction has to be performed with all other methods. Then, methods (Mohammadi & Vinciarelli, 2012; Schuller *et al.*, 2012; Eyben *et al.*, 2016) need to perform various operations like pitch extraction, power ratios, peak detection, linear regression, Viterbi-based smoothing, etc. In contrast, the proposed method needs to solve an optimization problem using the LARS-lasso algorithm which has the same computational complexity as regular least-square regression (Efron *et al.*, 2004). The fastest model at test time is SAE because it only needs to perform weight matrix multiplication to obtain the patch representation. Finally, one could argue that more memory is required with the proposed method as it needs to store the dictionary. However, a 800 word dictionary of 16×16 *pixel* patches require storing around 1.6 MB when using the double-precision floating-point format, which is highly manageable in modern computers.

3.6 Conclusion

This paper presents a new method for automated assessment of personality traits in speech. Speech segments are represented using spectrograms and feature learning. The proposed representation is compact and is obtained using a single algorithm requiring minimal expert intervention, when compared to reference methods. Experiments conducted on SSPNet corpus indicate that the proposed method yields the same level of accuracy as state-of-the-art methods in paralinguistics that employ more complex representations, while remaining simpler to use.

As explained in Section 3.5.1, the method is not properly equipped to capture pitch and speech rate. Research should be conducted to include these signal characteristics in the representation. In addition, experiments on different paralinguistic problems should be conducted to validate the applicability of the proposed method in different contexts. Experiments should also be conducted where the sparse dictionary learning and classifier algorithms used in our implementation is replaced by other methods enforcing group sparsity and discrimination. Finally, given the unsupervised nature of the feature learning process, experiments should be conducted

to assess the potential benefits of using a larger number of examples from other speech data sets.

CHAPTER 4

BAG-LEVEL AGGREGATION FOR MULTIPLE INSTANCE ACTIVE LEARNING IN INSTANCE CLASSIFICATION PROBLEMS

Marc-André Carbonneau[1,2], Eric Granger[1], Ghyslain Gagnon[2]

[1] Laboratory for Imagery, Vision and Artificial Intelligence,
École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] Communications and Microelectronic Integration Laboratory,
École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article submitted to « IEEE Transactions on Neural Networks and Learning Systems » in October 2017.

Abstract

A growing number of applications (e.g. video surveillance and medical image analysis) require training recognition systems from large amounts of weakly annotated data while some targeted interactions with a domain expert are allowed to improve the training process. In such cases, active learning (AL) can reduce labeling costs for training a classifier by querying the expert to provide the labels of most informative instances. This paper focuses on AL methods for instance classification problems in multiple instance learning (MIL), where data is arranged into sets, called bags, that are weakly labeled. Most AL methods focus on single instance learning problems. These methods are not suitable for MIL problems because they cannot account for the bag structure of the data. In this paper, new methods for bag-level aggregation of instance informativeness are proposed for multiple instance active learning (MIAL): The *aggregated informativeness* method identifies the most informative instances based on classifier uncertainty, and queries bags incorporating the most information. The other proposed method, called *cluster-based aggregative sampling*, clusters data hierarchically in the instance space. The informativeness of instances is assessed by considering bag labels, inferred instance labels and the proportion of labels left to discover in clusters. These proposed methods significantly

outperform reference methods in extensive experiments using benchmark data from several application domains. Results indicate that using appropriate strategies in MIAL problem leads to a significant reduction in the number of queries needed to achieve the same level of performance as single instance AL methods.

4.1 Introduction

years have witnessed substantial advances of machine learning techniques that promise to address many complex large-scale problems that were previously thought intractable. However, in many applications, annotating enough representative training data to train a recognition system is costly, and in such cases, one can resort to AL to reduce the annotation burden (Freund *et al.*, 1997; Dasgupta, 2011). Moreover, several applications allow to leverage some targeted interactions with human experts, as needed, to label informative data and drive the training process. AL has been used in various applications to reduce the cost of annotations, e.g., in medical image segmentation (Konyushkova *et al.*, 2015), text classification (Tong & Koller, 2001; Hoi *et al.*, 2006) and visual object detection (Vijayanarasimhan & Grauman, 2014).

Alternatively, the cost of annotations can be reduced through weakly supervised learning. It generalizes many kinds of learning paradigms including semi-supervised learning and MIL in partially observable environments or learning from uncertain labels. With MIL, training instances are grouped in sets (commonly referred to as bags), and a label is only provided for an entire set, but not for each individual instance. MIL has also been shown to efficiently reduce annotation costs in several applications such as object detection (where labels are obtained for whole images) (Ren *et al.*, 2016), description sentences (Xu *et al.*, 2016; Karpathy & Fei-Fei, 2015; Fang *et al.*, 2015) and web search engine results (Zhu *et al.*, 2015). This is particularly attractive for medical image analysis where a system can learn using labeled images that were not locally annotated by experts (Quellec *et al.*, 2017). Other successful applications of MIL include text classification (Ray & Craven, 2005; Zhang *et al.*, 2013), sentiment analysis (Kotzias *et al.*, 2015), and sound classification (Briggs *et al.*, 2012).

This paper focuses on methods that are suitable for MIAL problems. Although several AL methods exist for single instance learning (Settles, 2009), only a handful of methods have been proposed to address MIAL problems (Meessen *et al.*, 2007; Settles *et al.*, 2008; Zhang *et al.*, 2010; Melendez *et al.*, 2016a). Single instance active learning (SIAL) methods are not suitable for MIL because: 1) in MIL, instances are grouped in sets or bags, and 2) training instances have weak labels. The arrangement of instances into bags gives rise to several different tasks, such as bag classification and instance classification which must be addressed differently (Carbonneau *et al.*, 2016a).

Different learning scenarios exist for active MIL (Settles *et al.*, 2008). In this paper, we focus on the scenario where the learner has a set of labeled bags at its disposal, and must predict the label of each individual instance. The learner can query the oracle to label the bag's content. The final objective is to uncover the true labels of the instances, which corresponds to the transduction setting described in (Garcia-Garcia & Williamson, 2011). Given instances that are correctly labeled, any classifier can be used in a supervised fashion to classify instances not belonging to the training set in an inductive setting (Garcia-Garcia & Williamson, 2011). To our knowledge, this scenario has never been studied in the literature. The few existing MIAL methods focus on bag classification (Meessen *et al.*, 2007; Settles *et al.*, 2008; Zhang *et al.*, 2010) or select groups of instances in a scenario where there is only one query round (Melendez *et al.*, 2016a).

The MIAL scenario that we address is relevant in several real-world problems. For example, in some computer-assisted diagnosis applications, classifier is trained to identify localized regions of organs or tissues afflicted by a given pathology. A classifier is typically trained using afflicted regions identified by an expert or a committee of experts, which is costly in terms of time and resources. This limits the quantity of available data for training. However, it is easier to obtain images along with a subject diagnosis as a weak label (bag label). In order to make better use of the experts, the MIAL learner identifies the subject whose local annotations would most improve the classifier. In this example, we believe that our learning scenario is more plausible than the second scenario where instances are queried individually. When experts are asked

to provide local annotations of afflicted tissues or organs, it makes more sense to provide an entire image (bag) of the patient rather than provide isolated regions (instances). In this kind of applications, it is important for the annotator to be aware of the context provided by the surroundings of the segment when assigning a label. A similar argument can be made for text classification where an instance can be a sentence or a paragraph. It is easier to provide an accurate label for individual parts with knowledge of the entire text.

Beyond the well-known difficulties associated with AL, MIL instance classification raise several challenges. First, leveraging the weak supervision provided by bag labels is challenging because it is not explicitly known how each instance relates to its bag label. Also, the fact that training instances are arranged in sets adds an extra layer of complexity regarding relations between training instances. Moreover, in MIL, instance classification is often associated with severe class imbalance problems. Finally, AL and weakly supervised learning are often used to reduce the annotation cost of large amount of data which calls for algorithms with low computational complexity. For cost-effective design of an instance classifier through MIL, an AL algorithm should:

- characterize uncertainty in the instance space – assess which regions of the instance space are most ambiguous to the classifier, and thus informative for design.
- identify the most informative bag for the learner given multiple regions of the instance space.
- leverage bag label information, from queried and non-queried bags. This is in contrast to traditional AL problems because in our context bag labels provide weak indication of the instance labels.

Two new MIAL methods are proposed in this paper for bag-level aggregation of instance informativeness, allowing to select the most informative bags to query, and then learn. The first method – *aggregated informativeness* (AGIN) – assesses the informativeness of each instance to compute the informativeness of bags. Informativeness is based on classifier uncertainty,

and instances near the decision boundaries are prioritized. The second method – *cluster-based aggregative sampling* (C-BAS) – characterizes clusters in the instance space by computing a criterion based on how much is known about the cluster composition and the level of conflict between bag and instance labels. The criterion enforces the exploration of the instance space and promotes queries in regions near the decision boundary. Moreover, the criterion discourages the learner from querying about instances for which the label can be inferred from bag labels. Extensive experiments have been conducted to assess the benefits of using both proposed methods in three application domains: text, image and sound classification.

The rest of the paper is organized as follows. The next section reviews the state-of-the-art in active MIL. Section 4.3 formalizes the active MIL problem and presents the two proposed methods. The experimental methodology is described in Section 4.4, and results are analyzed and discussed in 4.5.

4.2 Multiple Instance Active Learning

This paper focuses on pool-based AL methods (Settles, 2009) where the learner is supplied with a collection of unlabeled and labeled samples. The learner must select the best instance, or groups of instances, to query. Pool-based AL problems have been tackled following two intuitions (Dasgupta, 2011): 1) queried instances should shrink the classifier hypothesis space as much as possible, and 2) cluster structure of the data should be exploited for efficient exploration of the input space. The methods proposed in this paper address the MIAL problem from each intuition perspective.

Several types of approaches shrink the classifier hypothesis space. The methods based on uncertainty query the most ambiguous instances for the classifier (Tong & Koller, 2001; Lewis & Gale, 1994) or the instance causing the most disagreement in a pool of classifiers (Seung *et al.*, 1992; Melville & Mooney, 2004). A drawback of these methods is that they tend to choose outliers for query since they are often ambiguous for the classifier (Tang *et al.*, 2002; Zhu *et al.*, 2008). To avoid this problem, some methods compute the expected error reduction (Roy & McCallum,

2001; Guo & Greiner, 2007) or expected model change (Settles *et al.*, 2008). They estimate the impact of obtaining each individual instance label on the generalization error or the model parameters. However, these methods are computationally expensive because classifiers must be trained for each possible label assignment of each unlabeled data sample. To avoid this problem, some methods aim to reduce generalization error by minimizing the model variance (Cohn *et al.*, 1994; Hoi *et al.*, 2006), typically by inverting a Fisher information matrix for each training instance. The size of the matrix depends on the number of parameters in the model which can rapidly become intractable (Settles, 2009). All these approaches are subject to sampling bias problems (Dasgupta, 2011), where some true instance labels may never be discovered for multi-modal distributions. This is because at the start of the learning process a classifier is trained using sampled data, and then later, queries are proposed near the decision boundaries of this classifier. If data structure exists, but was not captured by the initial samples, it may never be discovered.

Another group of AL methods relies on the characterization of the data distribution in the input space (Settles & Craven, 2008; Fujii *et al.*, 1998; Nguyen & Smeulders, 2004). Instead of concentrating on decision boundaries, they assess the structure of input data in order to query for informative instances that are representative of the input distribution. Leveraging the input data structure promotes exploration and discourages the selection of outliers. As a result, methods characterizing the input space yield better performance than other types of method when the quantity of labeled data is limited. However, as more labels are queried, methods that shrink the hypothesis space tend to perform better (Wang & Ye, 2015). The complexity of these approaches is generally similar to other kind of approaches with an added initial cost of a clustering or density estimation step (Settles & Craven, 2008).

As will be described in Section 4.3, the AL methods proposed in this paper follow these two different intuitions. AGIN seeks to shrink the hypothesis space based on classifier uncertainty, while C-BAS characterizes the data distribution. These methods have been developed with computational efficiency in mind, which is increasingly important to address the growing complexity of large-scaled applications.

Although MIL methods were initially proposed for bag classification (Amores, 2013), instance classification problems have more recently attracted growing interest (Vanwinckelen *et al.*, 2015; Vezhnevets & Buhmann, 2010; Xu *et al.*, 2016; Zhu *et al.*, 2015). These are different tasks that require different approaches (Carbonneau *et al.*, 2016a; Vanwinckelen *et al.*, 2015). MIL methods fall into one of two main categories depending on which level, bag or instance, discriminant information is extracted (Amores, 2013). Bag-level methods compare bags directly using set distance metrics or embed bags in a single summarizing feature vector (Chen *et al.*, 2006; Wang & Zucker, 2000; Cheplygina *et al.*, 2015a; Gärtner *et al.*, 2002; Zhou *et al.*, 2009). These methods do not perform instance classification and are unsuitable in our context. In contrast, instance-level methods predict the class of instances and combine these predictions to infer the bag label (e.g., APR (Dietterich *et al.*, 1997), DD and EM-DD (Maron & Lozano-Pérez, 1998; Zhang & Goldman, 2001), mi-SVM and MI-SVM (Andrews *et al.*, 2002), RSIS (Carbonneau *et al.*, 2016e) and MI-Boost (Babenko *et al.*, 2008)). While these methods are usually designed for bag classification, they can be employed for instance classification tasks. It has been shown that bag classification and instance classification tasks have different misclassification costs (Carbonneau *et al.*, 2016a), which means that the best bag classifier is not necessarily the best instance classifier (Vanwinckelen *et al.*, 2015). Moreover, experiments in (Carbonneau *et al.*, 2016a; Ray & Craven, 2005) show that single instance classifiers often perform comparably to MIL methods, especially for instance classification.

The literature on MIAL is limited and almost each method is proposed for a specific learning scenario. There are methods that query bag labels for bag classification. The method in (Meessen *et al.*, 2007) embeds bags in a single feature vector using a representation based on MILES (Chen *et al.*, 2006). An SVM is used for classification and the embedded bags which are closest to the decision hyper-plane are selected as in (Tong & Koller, 2001). This method has been generalized in (Zhang *et al.*, 2010) and a selection method based on Fisher's Information criterion has also been proposed. The learning scenario in (Settles *et al.*, 2008) is similar to ours in that all bag labels are known and the learner queries instance labels from positive bags. However, our goal is to train an instance classifier (not a bag classifier), and the learner queries

all instance labels from a bag (instead of only one), which we believe to be more efficient in practice. They train a logistic regression classifier optimized for bag-level classification accuracy. Their selection method is based on uncertainty sampling and expected gradient length. Queried instances are duplicated and added to the training set as singleton bags. While this method works well in practice, it is computationally expensive and the expected gradient length method is sensitive to feature scale (Settles, 2009). The method proposed in (Melendez *et al.*, 2016a) targets the instance classification task in a peculiar MIAL scenario where there is only one query round. First, instances are classified using a MIL algorithm (Melendez *et al.*, 2015a) and then, the most valuable instances are grouped in regions. These hundreds of regions are then labeled by an expert and the MIL classifier is retrained. This differs from the scenario in this paper because there is only one query round, and the expert must annotate a region instead of an image.

4.3 Proposed Methods

Figure 4.1 presents an overview of the MIAL framework for our learning scenario. The training data set $\mathcal{B} = \{B_1, B_2, ..., B_Z\}$ is a set of Z bags, each one is associated with a label $Y_i \in \{-1, +1\}$ and contains N_i instances: $B_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, ..., \mathbf{x}_{iN_i}, \}$. Each instance $\mathbf{x}_{ij}$ has an associated label $y_{ij} \in \{-1, +1\}$. All the bag labels are known a priori. Following the standard MIL assumption (Dietterich *et al.*, 1997), the labels of instances in negative bags are assumed to be negative, while positive bags contain negative instances and at least one positive instance:

$$Y_i = \begin{cases} +1 & \text{if} \quad \exists y \in B_i : \ y_{ij} = +1; \\ -1 & \text{if} \quad \forall y \in B_i : \ y_{ij} = -1. \end{cases} \tag{4.1}$$

The task consists in training a classifier to correctly predict the label of each individual instance $f(\mathbf{x}) \rightarrow y$. The classifier's decision function can be iteratively improved by querying an oracle about a bag. To select the most informative bag for query, the function $g(B) \rightarrow \mathbb{R}_{\geq 0}$ assigns an informativeness score to each of them. Once a bag has been selected for query (B^*), the oracle

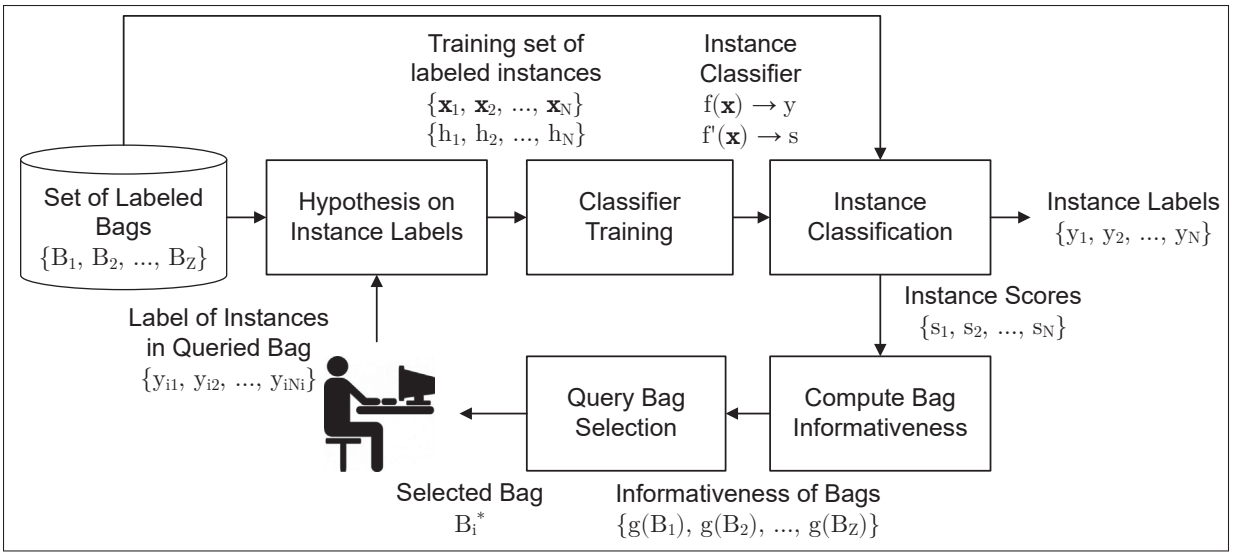


Figure 4.1 Block diagram of the general operations performed in our MIAL scenario for instance classification. The learner is initially supplied with a set of labeled bags, but no instance label. During each iteration, the learner predicts a label for each instance. An instance classifier is then trained, and used to assign a label and a score to all instances in the training set. The score of each instance is used to identify the most informative bag to query. Finally, the labels of all instances in the selected bag are annotated by the oracle in order to update the hypothesis and retrain the classifier.

provides labels for all its instances. Then, the hypothesis on instance labels h_{ij} is updated, and the classifier is retrained. The next best candidate bag for query is selected, and so on. The rest of this section presents two new methods to derive $g(B)$ for selecting bags for query.

4.3.1 Aggregated Informativeness (AGIN)

This method is inspired from SIAL methods (like in (Tong & Koller, 2001)) that select the instance expected to provide the largest reduction in the set of all consistent hypotheses. For instance, when working with SVM classifiers, this amounts to selecting the instance which is the closest to the decision hyper-plane. However, in MIL problems, instances are grouped into bags and the bag containing the single most informative instance is not necessarily the optimal choice. If the most informative instance is part of a bag containing only trivial instances, it may be advantageous to select another bag containing several difficult instances, even if none

of them are the single most informative instance in the entire data set. In other words, a bag should be selected based on the combined informativeness of its instances.

Here we describe the method as an adaptation of (Tong & Koller, 2001). The SVM classifier is used as an example, but it can easily be replaced with any type of classifier. First, the distance to the decision hyper-plane must be transformed into instance informativeness. Let $f'(\mathbf{x}) \rightarrow s$ be a function returning a classification score $s \in \mathbb{R}$ for an instance $\mathbf{x}$. This is the same as the classifier function $f(\mathbf{x})$, without a decision threshold.

For an SVM, the decision hyper-plane is defined by $f'(\mathbf{x}) = 0$. The informativeness of an instance can be obtained using a radial basis function $\phi(\mathbf{x})$ centered at 0. Any type of function can be used as long as it is maximized at the decision threshold, and it decreases monotonically with distance. In this paper we use:

$$\phi(\mathbf{x}) = e^{-2|f'(\mathbf{x})|} \tag{4.2}$$

This function decreases exponentially as the magnitude of s increases. This ensures that instances located close to the hyper-plane are highly prioritized over other less ambiguous instances.

The informativeness score of a bag is the aggregation of informativeness scores over all its instances:

$$g(B) = \sum_{\mathbf{x} \in B} \phi(\mathbf{x}) \tag{4.3}$$

The bag (B^*) with the highest informativeness score is selected for query:

$$B^* = \underset{B \in \mathcal{B}}{\operatorname{argmax}}\, g(B) \tag{4.4}$$

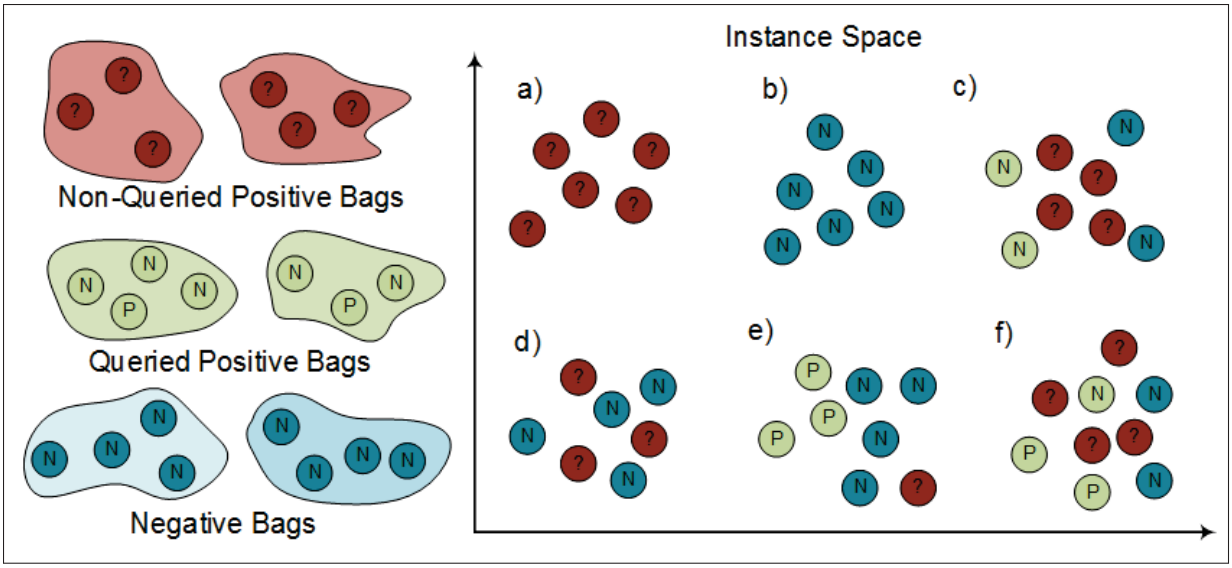


Figure 4.2 Representation of clusters in the instance space in an MIAL problem. It shows different types of cluster. In cluster a), even if none of the instance have been queried, they are considered non-informative because they all belong to bags of the same class. The same can be said about instances in cluster b). In cluster c) and d), all labeled instances belong to the same class even if their bag labels are different. The remaining instances are therefore deemed to be uninformative. Most of the instance labels in cluster e) are known and thus, the label of the remaining instance is unlikely to provide useful information. Instances in cluster f) should be informative because there is label disagreement at bag and instance level, and an appreciable proportion of instance labels remain to be discovered.

4.3.2 Clustering-Based Aggregative Sampling (C-BAS)

This method is proposed to alleviate problems associated with the sample bias, and to leverage the weak information provided by bag labels and classifier predictions on instance labels. The intuition behind C-BAS is that a cluster of instances should meet three conditions to be informative: 1) bag label disagreement, 2) instance label disagreement, and 3) contain a considerable proportion of non-queried labels. If a cluster contains instances from only one class of bags, the label of these instances is the same as the label of their bag. Obtaining the true labels for these instances is not informative. Inversely, if a cluster contains different types of instances, it should define a decision boundary. Acquiring labels in this cluster is likely to help

refine the overall decision boundary. Finally, to encourage exploration, clusters for which few labels are known will be considered as informative. Figure 4.2 illustrates these situations.

C-BAS starts by hierarchical clustering of data in the instance space. As in (Dasgupta & Hsu, 2008), we employ agglomerative hierarchical clustering, although it can be replaced with any type of hierarchical clustering algorithm. This type of method does not require setting the number of expected clusters a priori, and creates a clustering dendogram or tree that is used to create space partitioning of different granularities. The informativeness of instances in each cluster k is evaluated by a criterion c_k that accounts for cluster composition of the cluster. The criterion is composed of 3 terms enforcing the aforementioned conditions of informativeness:

$$c_k = BD_k \cdot ID_k \cdot E_k \tag{4.5}$$

The BD_k term measures the level of disagreement between bag labels with an entropy-based function:

$$BD_k = \frac{\beta\log(\beta) + (1-\beta)\log(1-\beta)}{\log(0.5)}, \tag{4.6}$$

where β is the proportion of instances from positive bags among the instances assigned to the cluster. If all instances come from bags of the same class, this term is equal to 0 which inhibits further research in this cluster. When bag labels are equally divided among the two classes, the term value is equal to 1. Similarly, the ID_k term measures the degree of disagreement between instance labels:

$$ID_k = \frac{\zeta\log(\zeta) + (1-\zeta)\log(1-\zeta)}{\log(0.5)}, \tag{4.7}$$

where ζ is the proportion of positive instances among the instances assigned to the cluster. When the true label of an instance remains unknown, the classifier's prediction is used as label. Finally, The term E promotes cluster exploration based on the proportion of unlabeled instances (α) in contains:

$$E_k = \frac{1-e^{-\alpha}}{1-e^{-1}}, \tag{4.8}$$

When all instance labels are known this terms is equal to 0, and when none are known, it is 1 .

Exploring the Clustering Tree

The clustering tree is explored from top to bottom. Iteratively the tree is pruned farther away from the trunk, each time yielding a clustering of finer granularity. For each clustering level $l \in \mathcal{L}$, the informativeness criterion c_k of each cluster k is computed. The informativeness $\phi(\mathbf{x})$ of an instance is an accumulation of the informativeness of each cluster k to which it was assigned:

$$\phi(\mathbf{x}) = \sum_{l \in \mathcal{L}} \sum_{k \in \mathcal{K}_l} \mathbb{1}_k(\mathbf{x}) \cdot c_k, \tag{4.9}$$

where $\mathcal{K}_l$ is the set of clusters obtained when the tree is cut at level l.

Different levels of granularity are necessary to correctly assess the informativeness of instances. By considering only large clusters obtained (top of the tree), all instances would be provide the same level of information. They would all be assigned to few large clusters which are likely to present a high level of disagreement between labels, and include many non-queried instances. Inversely, by considering very fine cluster granularity (bottom of the tree), the levels of disagreement between labels BD_k and ID_k tend towards 0, which means $c_k = 0$ and thus $\phi(\mathbf{x}) = 0$ for all $\mathbf{x}$. This is equivalent to randomly picking any unlabeled instances. Accumulating evidences on informativeness over levels of cluster granularity allows to compromise between the two extreme cases. Once all instance informativeness scores $\phi(\mathbf{x})$ are computed, the query bag B^* is selected in the same way as for AGIN (see (4.3) and (4.4)).

4.4 Experiments

All experiments were repeated 100 times and conducted with the following protocol. The data sets were randomly split in test (1/3) and training (2/3) subsets. For fair comparison, all MIAL methods are the same except for the bag selection scheme. The initial hypothesis for the labels individual instance is that they inherit the label of their bag, which is often successful in practice (Ray & Craven, 2005; Carbonneau *et al.*, 2016a). Bags are queried one by one until there are no positive bags left to query in the training set. After each query, the performance

of classifiers is measured on the training and test subsets. This corresponds to the transductive and inductive learning settings described in (Garcia-Garcia & Williamson, 2011).

As bags are queried, class imbalance of instance labels grows, which is an important concern for MIL instance classification tasks (Herrera *et al.*, 2016b). This is particularly true in data sets where the proportion of positive instances in positive bags is low. We handle class imbalance using Different Error Costs SVM (DEC-SVM) (Veropoulos *et al.*, 1999). This SVM method assigns different misclassification costs C to different classes. Table 4.1 reports the configuration of the SVM used for each data set. These parameters were obtained with 5-fold cross-validation using the real instance labels. We used the LIBSVM implementation (Chang & Lin, 2011). The ratio between the misclassification penalty cost of the classes corresponds to the class imbalance ratio ($\rho = \frac{N_+}{N_-}$). N_+ and N_- are the number of positive and negative instances in the training set. Each time an SVM is trained, class imbalance ratio is recomputed and misclassification costs are adjusted accordingly.

Performance is reported in terms of F_1-Score and the area under the precision-recall curve (AUC_{PR}) which are appropriate metrics for problems with class imbalance.

Table 4.1 SVM parameter configuration used in experiments

Dataset	$\mathbf{C}_+$	$\mathbf{C}_-$	**kernel**	γ
SIVAL	1000	ρ1000	Gaussian RBF	0.01
Birds	1000	ρ1000	Gaussian RBF	0.1
Newsgroups	1000	ρ1000	χ^2	-

To assess the benefits of employing bag selection schemes for query selection, the first reference method selects bags at random. It selects only positive bags since the label of instances in negative bags are assumed to be known. The few MIAL methods proposed in literature were not designed for instance classification, so the simple margin method (Tong & Koller, 2001) was considered as the second reference method. It consists in picking the closest unlabeled instance to the decision hyper-plane of the SVM. In our experiments the method selects the bag containing this most informative instance. This method is originally intended for single

instance learning scenarios and is closely related to AGIN. It is therefore relevant to show the effect of the proposed aggregation schemes.

4.4.1 Data Sets

The MIAL methods are evaluated using the three most widely used collection of MIL data sets providing instance annotations: Birds (Briggs *et al.*, 2012), SIVAL and Newsgroups. The last two were introduced to compare MIAL methods in (Settles *et al.*, 2008). They represent 3 different application domains – content-based image retrieval, text and sound classification. Each dataset contains different classes which are in turn used as the positive class yielding a total of 58 different problems. Table 4.2 gives an overview of the properties for each data set.

Table 4.2 Summary of the properties of the benchmark data sets

					Inst. per Bag			**Class imbalance**		
Name	**Sets**	**Bags**	**Inst.**	**Feat.**	Min.	Max.	Avg.	Min.	Max.	Avg.
SIVAL	25	180	5690	30	31	32	32	0.035	0.218	0.095
Birds	13	548	10232	38	2	43	19	0.003	0.143	0.040
Newsgroups	20	100	4060	200	8	84	40	0.012	0.035	0.018

4.4.1.1 SIVAL

The Spatially Independent, Variable Area and Lighting (SIVAL) data set for visual object retrieval (Rahmani *et al.*, 2005) contains 1500 images each depicting one of 25 complex objects photographed from different viewpoints in various environments. The version used in this paper has been segmented and hand-labeled to compare MIAL approaches in (Settles *et al.*, 2008). Each object is in turn considered as the positive class, and all remaining objects are part of the negative class. This yields 25 different 2-class learning problems. Each of the 25 data sets contains 60 positive images and 120 negative images sampled uniformly from all 24 negative classes. Images are represented as bags which are a collection of segments. Texture and color features are extracted from segments as well as neighborhood information yielding a 30-dimensional feature vector for each. The proportion of positive instances in positive bags

is 25.5% in average and ranges from 3.1% to 90.6%. This data set exhibits high intra-class variation which means that the positive instance distribution is multimodal.

4.4.1.2 Birds

This data set (Briggs *et al.*, 2012) contains recordings of bird songs captured by unattended microphones in the forest. Each bag is the spectrogram of a 10 seconds recording. The recording is temporally segmented and 38 features characterizing shape, time and frequency profile statistics are extracted from each segment. The data set contains 13 species of birds, which are in turn considered as the positive class yielding 13 problems. This data set is difficult because in some cases there is extreme class imbalance at bag and instance level. For example, there are only 32 instances out of 10232 that belong to the hermit thrush. In the best case, positive instances represent 12.5% of all instances. As opposed to the other data sets, each class (except for background noise) is represented by a single compact cluster in space.

4.4.1.3 Newsgroups

This MIL data set was created using instances from the *20 Newsgroups* data set corpus in (Settles *et al.*, 2008). Instances are posts from newsgroups about 20 different subjects. Each post is represented by a 200 term frequency-inverse document frequency feature vector. For each version of the data set, a subject is selected as the positive class and the remaining 19 other subjects constitute the negative class. A bag is a collection of posts. The feature vectors used for this data set are sparse histograms which makes the distribution different from the two other problems. It constitutes a good way to evaluate the robustness of the proposed method to different data distribution types. Moreover, the average proportion of positive instances in positive bags is rather low, which also makes the problem difficult and accentuate problems related to class imbalance.

4.4.2 Implementation Details for C-BAS

Here we detail the particular implementation of C-BAS that we use in the experiments. The clustering three is obtained using the Ward's average linkage algorithm. We then obtain different clustering refinements by cutting the tree at different levels. To make sure to cut at significant levels in the tree, we compute the inconsistency coefficient δ of all links in the tree:

$$\delta_k = \frac{h_k - \mu_{\mathcal{N}_k}}{\sigma_{\mathcal{N}_k}}, \tag{4.10}$$

where h_k is the height of the link k (cophenetic distance between the clusters). The set $\mathcal{N}_k$ contains all links in the P hierarchical levels under k. $\mu_{\mathcal{N}_k}$ and $\sigma_{\mathcal{N}_k}$ are the average and the standard deviation of the height of the links contained in $\mathcal{N}_k$. A high inconsistency coefficient means that the two clusters joined by the link are farther apart then the clusters linked in the levels below, which indicates a natural separation in the data structure.

Once the inconsistency coefficients δ has been computed for all links, they are sorted from highest to lowest. Clusters are obtained using these values as thresholds. Instances or clusters can only be linked together if the inconsistency coefficient of the link is lower than the threshold. Iteratively, the threshold is lowered and finer clusterings are obtained. In the experiments of this paper, we use 20 threshold levels and P has been arbitrarily set to 16 for all data sets. Both parameters could be optimized depending on the application.

4.5 Results and Discussion

MIAL methods are evaluated based on their ability to uncover the true instance labels in the training set (transductive learning task) and to classify a test set with a classifier trained using these uncovered labels (inductive learning task). Fig. 4.3 shows an example (over 100 runs) of the evolution of average F_1-score values on the training subset as a function of the number queries to the oracle. Similar learning curves were obtained with AUC_{PR} but are not shown here since they do not provide pertinent additional information. Results show that for each data set, the proposed methods can significantly improve the learning process. Each curve starts (no

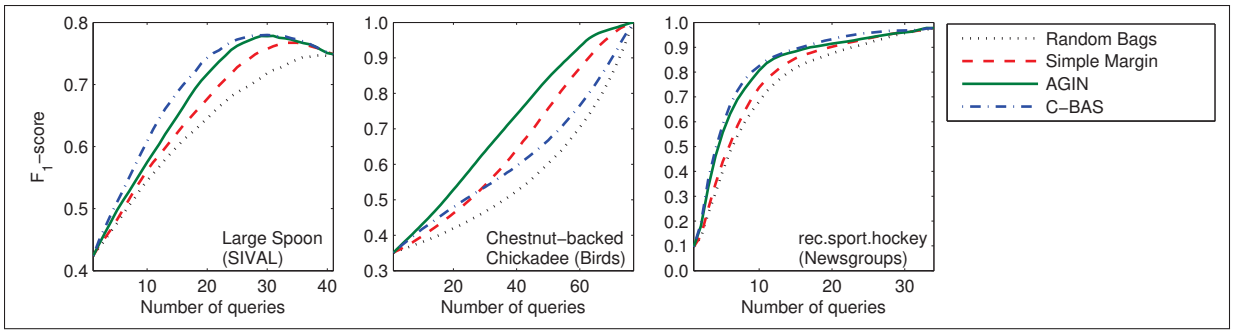


Figure 4.3 Average learning curves for MIAL methods on SIVAL, Birds and Newsgroups datasets

bags have been queried) and finishes (all true instance labels are known) at the same level of performance.

From these curves, it is possible to see how many queries are necessary to achieve the same level of performance with different methods. For example, selecting random bag may necessitate as much as 23 (out of 40) more queries than C-BASS to obtain the same F_1-Score on the Glaze Wood Pot training set. This is a best case scenario but nonetheless, out of the 58 data sets, using AGIN has lead to a reduction of the number of query necessary on all but 1 test data set with the AUC_{PR} metric. Similarly, C-BASS has resulted in a query reduction for all but 2 data sets.

In some of these curves, after a certain number of queries, the performance starts to decrease (see Fig. 4.3). While it seems counter-intuitive, this can be explained by the fact that the metric reported in the graph is different from the surrogate loss function used as an optimization objective. In our case, the SVM optimizes the hinge loss over all instances which does not guarantees the optimization of the F_1-Score (see (Loog & Duin, 2012; Loog *et al.*, 2017) for a more detailed discussion on the subject).

To compare the overall performance of methods for the entire AL sequence, the normalized area under the learning curve (NAULC) was used for both F_1-score and AUC_{PR} metrics. It corresponds to the area under curves as displayed in Fig. 4.3 divided by the total number of queries. For each problem in each data set, we compute the average NAULC and identify the

best performing method as a win. Statistical significance of results is assessed using a t-test (α=5%). Table 4.3 reports the number of wins for all methods (complete result tables can be found in the supplementary material document). Both proposed methods outperform the reference methods for all three application domains and for both the transductive and inductive tasks. Results indicate that aggregating instance informativeness to select queried bags is a better strategy than selecting the most ambiguous instance, and that SIAL methods should be adapted to MIL problems to improve performance.

Results suggest that proposed methods are better suited for different type of data. For example, AGIN outperforms other methods on the Birds dataset, while C-BAS yields better results with SIVAL data. Indeed, the positive instances in Birds data are likely to be grouped in very few clusters since birds of the same specie tend to have similar songs. In that case, the best strategy is to concentrate on refining the decision boundary since there are no hidden cluster structure to discover. Inversely, the positive distribution in SIVAL data is likely to have several modes. The appearance of an object, and thus its corresponding feature representation, can be very different depending on point-of-view, scale and illumination. In that case, it is important to discover these multiple clusters as rapidly as possible, which favors the C-BAS approach.

Table 4.3 Number of wins for each algorithm on each corpus. The NAULC for 100 runs were averaged and a t-test was performed to determine the best algorithm ($\alpha = 0.05$)

		Random Bags		**Simple Margin**		**AGIN**		**C-BAS**	
Task Setting	**Dataset**	F_1	AUC_{PR}	F_1	AUC_{PR}	F_1	AUC_{PR}	F_1	AUC_{PR}
Transductive	SIVAL	0	0	2	3	14	7	**19**	**23**
(Training set)	Birds	0	1	0	1	**13**	**12**	2	8
	Newsgroups	0	0	1	2	8	16	**19**	**17**
	TOTAL WINS	0	1	3	6	35	35	**40**	**48**
Inductive	SIVAL	3	1	13	12	**21**	**20**	18	19
(Test set)	Birds	2	5	3	4	**13**	**12**	8	**12**
	Newsgroups	6	6	10	17	**20**	**20**	18	16
	TOTAL WINS	11	12	26	33	**54**	**52**	44	47

The results in Table 4.3 suggest that AGIN and C-BAS are better suited for different tasks. This is because uncovering the labels of instances in labeled bags is slightly different than training a

classifier that generalizes well to unseen data. This has to do with how the algorithms approach the problem, class imbalance and the initial hypothesis on instance labels. The initial hypothesis that all instances inherit their bag labels ensures that all positive instances are used for training the classifier. At the same time, many negative instances are falsely labeled positive (FP). These noisy labels do not necessarily pose a serious difficulty when training the classifier. In regions densely populated with negative instances, FP are outweighed by true negative instances, and thus, overlooked by the classifier. In regions where there is a mix of true positives and negatives, FPs artificially expand the classifier positive regions which has the effect of increasing the sensitivity of the classifier. This means that, as bags are queried, precision increases but recall decreases. The initial increased sensitivity of the classifier has a beneficial effect on generalization (under these metrics) in context where there is class imbalance. Therefore preserving this effect while refining the decision boundary insures better generalization while learning. This explains why AGIN performs better for test set classification. Inversely, C-BAS uncover FP in all regions of the instance space which helps in yielding better results for the transductive task but mitigates the beneficial effect of the temporary increased sensitivity when compared to AGIN.

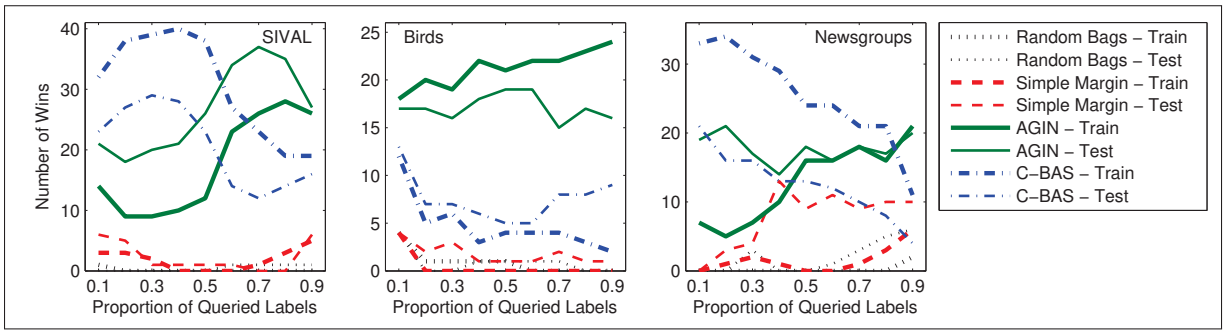


Figure 4.4 The number of wins of each method (both metrics) vs. the proportion of queried bag labels.

It had been previously shown that when very few instances are labeled, methods characterizing the distribution of the input space, like C-BAS, perform better than methods reducing the classifier hypothesis space, like AGIN, and vice-versa (Wang & Ye, 2015). This is observed in our

experiments (see Fig. 4.4). This is because C-BAS pushes the learner to quickly explore the most promising data clusters through the *E* term. Moreover, the *BD* term prevents the learner from querying instance labels that can be inferred from bag labels. After a certain number of queries, it becomes more important to refine decision boundaries, and that is when AGIN performs better.

For instance classification problems in MIAL, the exploration of the instance space is always promoted indirectly, which reduces the severity of sample-bias problems as found in SIAL problems. This implicit exploration comes from the fact that all instances of a queried bag are labeled together. Even if a bag is selected because it contains instances near a decision boundary, the other instances in the bag provide information about other regions of the instance space. This helps AGIN achieve a high level of performance. Based on these experiments, it seems that the AGIN method is preferable to the others in many situations. It achieves a high level of accuracy while remaining fairly simple to implement. It exhibits competitive levels of performance in both transductive and inductive learning tasks. There are two situations where it is preferable to use C-BAS: 1) when there are few known labels, and 2) when the positive instances are distributed in several regions of the input space.

While the proposed methods perform well with the type of data used in our experiments, we believe that there are some types of MIL problems were they might not yield optimal performance. As explained in (Carbonneau *et al.*, 2016a) MIL problems can possess several characteristics which require special care. Some of them would probably be difficult to address with the proposed algorithms. For example the proposed methods assume that all features are relevant for classification. This makes it difficult to deal with MIL data presenting strong intra-bag similarity. This means that instances from the same bag are similar and thus located in the same region of space. Also, AGIN and C-BASS were developed under the standard MIL assumption where all instances in negative bags are assumed to be negative. This assumption is sometimes violated in practice. Finally, the algorithms are designed for single bag query. In batch mode AL contexts the oracle is ask to label a set of query. The proposed algorithms

do not implement a mechanism that ensure that bags contained in a set of query are different, which might be sub-optimal in this context.

4.6 Conclusion

This paper introduces two methods for MIAL in instance classification problems. Experiments show that leveraging the bag-level structure of data provides a significant reduction in the number of queries needed to accurate classifiers for difference benchmark problems. Future research includes studying how different types of structure and correlation within and between bags affect the behavior of MIAL algorithms. An extension of the methods should be proposed mitigate the effect of similar instance in a same bag and to improve the batch mode learning process. Finally, experiments will be conducted to measure the benefit of using MIAL on data collected from large real-world clinical contexts.

CONCLUSION AND RECOMMENDATIONS

This thesis brought several contributions to MIL, from various angles, with a constant focus on the applicability to real-world scenarios. Throughout the research, the MIL algorithms were analyzed, developed and benchmarked with considerations for versatility, implementation cost and effort. Guidelines were given for practitioners for adequate use of MIL techniques, given application types.

It was first shown that training a classifier from MI data poses several challenges. The ambiguity on instance labels makes it difficult reliably train a classifier. Sometimes, instances do not have definite labels. The arrangement of instances in bags gives rise to relations of various natures such as co-occurrence and intra-bag similarities. The bag structure of MIL problems cannot be neglected when dealing with these relations. This is true for instance- and bag-level classification and in active learning frameworks. All of these relations and data characteristics have been rigorously surveyed and studied in the first chapter of this thesis. Application domains of MIL were discuss in regard of these characteristics. Extensive experiments were conducted to compare the behavior of a wide array of MIL approaches when facing data with challenging characteristics. Best performing types of approaches were identified for each case. The paper ends on a discussion on experimental protocols and open challenges for MIL. The main conclusions from the experiments were that:

- For all methods, a lower WR translates into lower accuracy;
- For the instance classification task, higher WR does not necessarily translates into higher accuracy (this conclusion relates to multimodal distributions);
- Supervised classifiers are as effective for instance classification as the best MIL classifiers when the WR is over 50%;
- In general, bag-space methods outperform their instance-space counterparts at higher WR;

- At lower WRs, there are other factors to consider when selecting a method (e.g. distribution shapes or intra-class variation);
- With most algorithms, performance decreases when the test negative instances distribution differs from the training distribution;
- The minimal Hausdroff distance is a powerful tool to deal with changing negative distributions;
- Score functions learned by the algorithms are still suitable when the negative distribution changes, but the thresholds should be adjusted;
- Embedding methods make no distinction between the positive and the negative class;
- Embedding strategies based on the characterization of instance distributions in bags are robust to noise;
- Instance-space methods are vulnerable to noise.

The rest of the thesis discussed MIL classification from different points of view, in different challenging contexts. First, a method was proposed to identify positive instances in MIL data sets. It projects instances into random subspaces and infers labels based on bag labels proportions in data clusters. Experiments show that the method outperform state-of-the-art reference methods in various conditions. The method was later used to build an ensemble of classifiers used for bag classification. State-of-the art results were obtained on several benchmark data sets. More importantly the method maintained high level of performance on a wide range of problems with different characteristics. The method has been shown to be robust to low WR, feature noise as well as being able to deal with many types of distributions. Based on these results one can conclude that cluster analysis in random subspaces is an efficient way to identify witnesses. Moreover, ensembling classifier proves to add robustness to the classification

process. Furthermore, it makes the method scalable and provides a way to deal with class imbalance at instance-level. However, the method works under the standard MIL assumption, and therefore, cannot deal with structured bags, co-occurrence and requires that the label space for instances is the same as for bags.

Then, a bag-level classification method was proposed for personality prediction from speech. The proposed method is a bag embedding method. Patches are extracted from spectrograms and used as instances. The fact that these instances cannot be assigned to a clear class makes the problem challenging. At first, a sparse coding algorithm learns important concepts in the data. Instances are later encoded as a composition of these concepts. Then encoded instances are embedded in a single feature vector representing the whole bag (i.e. speech signal). Experiments show that the method achieves state-of-the-art results while being simpler to implement than commonly used methods for this application. This chapter showed that MIL classifier are useful tools when learning from composite objects and that they can be used in cases where it is not possible for a human annotator to identify the discriminative part of such objects. Moreover, results indicate that using soft concepts assignment is a powerful strategy to describe instances. From the application point of view there were some paralinguistic cues that could not be leveraged by the representation method. Since the method relies on local patches, long term information such as speech rate and pitch variation cannot be encoded.

Finally, two methods were proposed to select the best bags to query in a MIL active learning scenario where the objective is to train an instance classifier. The first method focuses on refining the classifier decision boundary, while the second does a characterization of the input space for efficient exploration. Experiments showed that it is important to consider the bag structure of the problem in MIAL. Both methods achieved better performance than the similar SI active learning method. As previously observed by other AL researchers, uncertainty sampling methods offer best performance when a larger quantity of data has been annotated.

This is particularly true in our MIAL scenario where exploration of the input space is indirectly promoted because all instances in a bag a labeled by the oracle after a query. This means that instances that were not deemed informative by the uncertainty sampling scheme are also indirectly queried which mitigates the sample-bias problem. Moreover, the experiments indicate that characterizing the input space is a better strategy in transductive learning scenarios, while uncertainty sampling is preferable in inductive learning settings. However, this hypothesis should be verified with larger experiments.

Overall, this thesis gave a better understanding of the characteristics that make MIL unique. These unique characteristics are associated with challenges which limit the performance of MIL methods is real-world problems. We proposed methods to address some of these challenges. Each of these methods was proposed for a specific task under the appropriate assumption. Experiments showed that the strategy proposed to address the challenges were reliable and helped give an understanding of MIL classification in general.

Future Work

The active learning query methods proposed in Chapter 4 were designed with a special application in mind. This learning scenario would be appropriate to reduce the cost of annotation in medical imaging applications. Instead of having clinicians locally annotate complete sets of images, the proposed algorithms could be used to select fewer, but more informative images for annotation. So far, the algorithms have been successfully applied to MIL benchmark data sets, and now they should be validated on real-world medical image data.

We should also explore other affective learning application for the method proposed in Chapter 3. For instance, it would be interesting to see how the method performs on emotion recognition. Also, in its current form, the same dictionary is used to encode different regions of the frequency spectrum. It might be possible to improve accuracy by using separate dictionaries

for the lower and higher ends of the spectrum because they contain information of different natures. The lower end of the spectrum carries pitch and intonation information while the formants are in the higher part of the spectrum.

Many of the comparative experiments conducted in the thesis showed that mi-SVM is one of the very best methods for instance classification. This method is initialized with the assumption that positive instances in positive bags are all positive. This initialization phase could benefit from other types of transductive algorithms used in semi-supervised problems, or the RSIS method proposed in Chapter 2.

In Chapters 1 and Annex II, it has been established that bag-level and instance-level classification have different cost functions. Methods should be proposed to attack the instance-level classification task more directly. Possibly with an energy function dependent on label assignations in which different cost terms would enforce correct bag classification, correct negative instance classification and that similar instances are assigned the same label.

ANNEX I

WITNESS IDENTIFICATION IN MULTIPLE INSTANCE LEARNING USING RANDOM SUBSPACES

Marc-André Carbonneau[1,2], Eric Granger[1], Ghyslain Gagnon[2]

[1] Laboratory for Imagery, Vision and Artificial Intelligence,
École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] Communications and Microelectronic Integration Laboratory,
École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article published in « Proceedings of the International Conference on Pattern Recognition »
in December 2016.

Abstract

Multiple instance learning (MIL) is a form of weakly-supervised learning where instances are organized in bags. A label is provided for bags, but not for instances. MIL literature typically focuses on the classification of bags seen as one object, or as a combination of their instances. In both cases, performance is generally measured using labels assigned to entire bags. In this paper, the MIL problem is formulated as a knowledge discovery task for which algorithms seek to discover the witnesses (i.e. identifying positive instances), using the weak supervision provided by bag labels. Some MIL methods are suitable for instance classification, but perform poorly in application where the witness rate is low, or when the positive class distribution is multimodal. A new method that clusters data projected in random subspaces is proposed to perform witness identification in these adverse settings. The proposed method is assessed on MIL data sets from three application domains, and compared to 7 reference MIL algorithms for the witness identification task. The proposed algorithm constantly ranks among the best methods in all experiments, while all other methods perform unevenly across data sets.

2. Introduction

In multiple instance learning problems, instances are grouped in bags, and a label is provided for the whole bags. The individual labels of the instances are unknown. The standard formulation of MIL assume negative bags do not contain positive instances, while positive bags are said to contain at least one positive instance, called witness (Amores, 2013).

MIL have been successfully applied to various applications, such as molecule conformation classification (Dietterich *et al.*, 1997) and content-based image retrieval (CBIR) (Andrews *et al.*, 2002; Li *et al.*, 2009; Chen *et al.*, 2006). More recently, MIL algorithms attracted attention in the medical community, especially for computer-aided diagnostic from images (Quellec *et al.*, 2016; Kandemir *et al.*, 2014a; Melendez *et al.*, 2015b) because it allows learning from loosely annotated images.

In some applications, phenomenons are quantified using a set of observations. Identifying the truly informative instances, the witnesses, helps researchers better understand the phenomenon. For example, Palachanis (Palachanis, 2014) uses MIL to identify the genomic features governing the bonding of transcription factors in gene expression. In this case, bags represent genes, and transcription factors are instances. Witnesses are identified, and found to be corresponding to biological observations. In automated personality assessment from speech signals, data sets are created by psychologists that assign personality traits labels to whole speech segments. These experts perform this task intuitively, and thus, it is not clear what parts of the signal provided relevant cues for classification (Mohammadi & Vinciarelli, 2012). Being able to identify witnesses from positive bags could provide insight on the nature of data. As another example, by comparing the social media posts that a user either reads or ignores, one could infer user-specific elements of interest. All these cases correspond to the identification of witnesses in MIL data sets, which is more of a knowledge discovery task than a classification task.

Not all MIL algorithms allow to classify instances instead of bags. Many MIL algorithms based on bag distance measures (Wang & Zucker, 2000; Cheplygina *et al.*, 2015a) and bag embedding (Bunescu & Mooney, 2007b; Zhou *et al.*, 2009) do not provide information at

instance-level, and therefore cannot directly be used in witness identification problems. However, some of these methods, like MILES (Chen *et al.*, 2006) and Citation-kNN (Zhou *et al.*, 2005b), can be adapted for the task. In contrast, instance-based MIL methods like axis parallel rectangle (APR) (Dietterich *et al.*, 1997), mi-SVM, MI-SVM (Andrews *et al.*, 2002) and KI-SVM (Li *et al.*, 2009) infer bag labels based on individual instance classification, and thus can be used directly for witness identification. Although these methods can achieve a high level of performance in specific situations, they often perform poorly when the proportion of positive instances in positive bags, hereafter called the witness rate (WR), is low. In other cases, the methods cannot deal with witnesses sampled from multimodal positive data distributions. The modes of the distributions are clusters corresponding to latent variables in the data set, which will hereafter be called concepts.

In this paper a new method named Random Subspace Witness Identification (RSWI) is proposed. A related method was used in (Carbonneau *et al.*, 2016e) to design MIL ensembles for classification, and was shown to be robust to both low WR and multi-concept problems. RSWI computes a score for each instance that corresponds to its likelihood of being a witness. To compute these scores, all instances of the data set are projected in several random subspaces. Clustering is performed in each subspace, and the proportion of instances belonging to positive bags in each cluster is computed. The score of an instance is obtained by adding these proportions for each cluster it was assigned to. The random subspaces help capture relations in the data and provide robustness against the effects of irrelevant and redundant features, especially when using distance-based clustering methods like k-means with Euclidean distance.

To validate RSWI, the performance of several MIL algorithms with witness identification capabilities are compared and analyzed. Since witness identification is an aspect that has not yet been deeply explored, most existing MIL data sets do not include instance-level annotation. Thus, 2 new data sets have been created using data from real-world applications. The data sets were made publicly available by the authors on his personal website (https://sites.google.com/site/marcandrecarbonneau/).

3. Witness Identification in MIL Methods

Several instance-based MIL methods have been proposed for MIL. Instance-based methods classify instances individually and then, using instance labels, infer the label of the bag. These methods are suitable for witness identification. However, classifying bags differs from classifying individual instances. For example, under the standard MIL assumption that a positive bag contains at least one positive instance, when classifying a bag, once a positive instance has been identified, false negatives have no impact. Therefore, the best bag classifier is not necessarily the best instance classifier (Vanwinckelen *et al.*, 2015). This section describes the witness identification strategy of several instance-based MIL methods.

The simplest approach, which is not a MIL method *per se*, is to consider that the label of each instance corresponds to the label of the bag it belongs to, and train a regular supervised classifier. The negative instances in positive bags add noise to the optimization process. If the proportion of noise is low, this method performs relatively well, but performances rapidly decrease when the WR is low.

One of the first MIL methods, APR (Dietterich *et al.*, 1997) searches for a hyper-rectangle in feature space containing mostly instances from positive bags, and as few as possible instances from negative bags. The instances the hyper-rectangle encompasses are considered to be witnesses. While this method is successful in some situations, it has problems dealing with multimodal positive data distributions.

Two of the first MIL methods based on SVMs, mi-SVM and MI-SVM were proposed in the same paper (Andrews *et al.*, 2002). Both methods intrinsically perform witness identification, but differ in the strategy used to discover witnesses. In mi-SVM, the margin is maximized jointly over the discriminant function and individual instance label assignations of the complete data set. At first, a label is assigned to each instance, and an SVM is trained based on the instance label attribution. Instances are then reclassified using the newly trained SVM. The resulting labels are then assigned to each instance and the SVM is retrained. This procedure is repeated until the labels are stable. The witnesses are the instances with a positive label. MI-

SVM uses the same iterative procedure, except that positive bags are represented by the single most positive instance in the bag. Because it selects only one instance in each bag, this method has problems dealing with bags containing positive instances from more than one concept.

Instead of looking for witnesses directly, Maron and Lorenzo-Pérez proposed a measure called diverse density (DD) (Maron & Lozano-Pérez, 1998). This measures the probability that a given point in feature space belongs to the positive class. It depends on the proportion of instances from positive and negative bags in the neighborhood. The highest point of the DD function corresponds to the positive concept from which are generated the witnesses, and instances are classified based on their proximity to this point. Later, in EM-DD (Zhang & Goldman, 2001), the Expectation-Maximization algorithm was used to locate the maximum of the DD function. Because these methods seek a single maximum point, they assume that positive instances come from a single compact cluster in feature space, which limits their applicability to many problems. It has also been pointed out that EM-DD performance decreases when the number of noisy features increases (Zhang & Goldman, 2001). DD and SVM are combined in DD-SVM (Chen & Wang, 2004). Local maxima of the DD function are selected and used as prototypes. The distances between the prototypes and the instances in bags are used as feature vectors, which are classified by an SVM. MILES (Chen *et al.*, 2006) uses the same kind of distance-based embedding except that the prototypes are replaced by instances selected from the data set using a 1-norm SVM. The authors provided a way to identify witness based on each instance contribution to the bag label.

Some methods were proposed specifically to locate regions of interest (ROI) in images for CBIR. For example, CkNN-ROI (Zhou *et al.*, 2005b) classifies bags using the Hausdorff distance and the reference and citations scheme of Citation-kNN (Wang & Zucker, 2000). Once a bag is deemed positive, each instance it contains is treated as a bag, and is classified individually. The instances classified as positive are the witnesses. KI-SVM (Li *et al.*, 2009) also locates ROI by finding the key instance (i.e. witness) in bags using multiple kernel learning. The program is constrained to correctly classify each instance in negative bags. In the MKL formulation, each possible instance label assignation in positive bags corresponds to a kernel.

The algorithm seeks a combination of kernels which produces a correct label assignment in the data set. During its optimization, the constraints are satisfied if the bags are correctly labeled, and thus, if the positive bags contain more than one witness from different concepts, some witnesses can be ignored.

Most of the methods are less effective when the WR is low, or when the data sets contain two or more positive concepts. The proposed algorithm, RSWI (see next Section), consistently provides a high level of performance; it is robust to a large range of WR and allows to learn from multi-concept distributions.

4. Random Subspace Witness Identification

In this paper a new method called RSWI is proposed. It identifies witnesses by analyzing the neighborhood composition of each instance. The neighborhoods are defined by clusters in multiple random subspaces. The method is related to DD in the sense that this is a measure of the likelihood that an instance is positive, but instead of locations in feature space, a score is given to instances. An advantage of RSWI is that there is no search for a global maximum, which makes the method robust to multimodal distributions. Moreover, RSWI performs a series of simple tasks which are computationally efficient. (see Figure I-1).

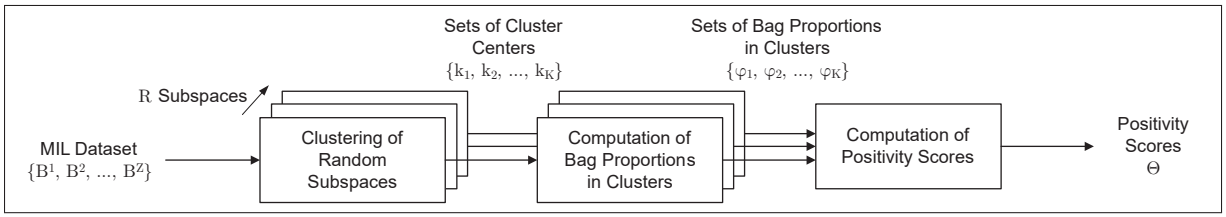


Figure-A I-1 Block diagram for positivity scores computation

In MIL problems $\mathcal{B} = \left\{B^1, ..., B^Z\right\}$ is a set of Z bags, each corresponding to a label $L^i \in \{-1, +1\}$. Each bag contains N^i couples composed of a feature vector and its associated label: $B^i = \left\{(\mathbf{x}^i_1, y^i_1), ..., (\mathbf{x}^i_{N^i}, y^i_{N^i})\right\}$ where $\mathbf{x}^i_j = (x^i_{j1}, ..., x^i_{jd}) \in \mathbb{R}^d$. The labels y^i_j of each individual instance are unknown in positive bags, but are assumed to be negative in negative bags. Fol-

lowing the standard MIL assumption (Amores, 2013), there is at least one positive instance per positive bag.

With RSWI, instances are identified based on a *positivity score* computed as follows: At first, subspaces $\mathcal{P}$ are created by randomly selecting p features from the complete set of d features. Every instance $\mathbf{x}$ in the data set is projected in the p-dimensional subspaces. Next, the data in each subspace is clustered. Here, a hard assignment method (e.g. k-means), is assumed, but any clustering algorithm could be used. Each subspace captures a different relation between instances resulting in different clusterings. The second step consists in computing the proportion φ_n of instances belonging to positive bags in each cluster k_n:

$$\varphi_n = \frac{\sum_{\forall \mathbf{x}} c(\mathbf{x}^i, n)}{|\mathcal{K}_n|} \in [0..1], \tag{A I-1}$$

where $n = 1, 2, ..., K$, and

$$c(\mathbf{x}^i, n) = \begin{cases} 1, & \text{if } \mathbf{x}^i \in \mathcal{K}_n \text{ and } L^i = +1; \\ 0, & \text{otherwise.} \end{cases} . \tag{A I-2}$$

In these equations, $\mathcal{K}_n$ represents the set of instances belonging to cluster k_n. The size of this set is given by $|\mathcal{K}_n|$.

These two steps (projection into a random subspace and clustering), are repeated R times. The third and last step is the computation of the instances positivity score $\theta(\mathbf{x})$. This score is the mean of all the positive bag proportion $\varphi_n(r)$ of the clusters it was assigned to:

$$\theta(\mathbf{x}) = \frac{1}{R} \sum_{r=1}^{R} \sum_{n=1}^{K} \varphi_n(r) \cdot d(\mathbf{x}, n, r), \tag{A I-3}$$

where

$$d(\mathbf{x}, n, r) = \begin{cases} 1, & \text{if } \mathbf{x} \in \mathcal{K}_n \text{ at repetion } r; \\ 0, & \text{otherwise.} \end{cases} \tag{A I-4}$$

These positivity scores give an indication of the likelihood that an instance is a witness. The label for $\mathbf{x}$ is given by:

$$y = \begin{cases} +1, & \text{if } \theta(\mathbf{x}) > \alpha; \\ -1, & \text{otherwise,} \end{cases} \tag{A I-5}$$

where α is the decision threshold. If labeled instances are available, this threshold should be optimized based on the desired performance measure. However, in most MIL problems, instances labels are unavailable. In that case, the threshold can be set by making sure at least one instance is classified as a witness in each positive bag:

$$\alpha = \min_{B_i \in \mathcal{B}^+} \left\{ \max_{\mathbf{x} \in B_i} \theta(\mathbf{x}) \right\} \tag{A I-6}$$

where $\mathcal{B}^+$ is the set containing only the positive bags. Following (A I-6), there will be at least one bag containing only one witness, but the other bags may contain any number.

Because RSWI is a local measure of positivity, it allows to identify witnesses in different regions of the feature space, making the algorithm robust to multimodal distributions. Also, since this measure is relative to all instances in the data set, witnesses can be identified reliably regardless of the WR.

5. Experimental Methodology

In many MIL papers, the accuracy is used as a performance metric. While reasonable when evaluating bag classification, it may be misleading in the context of instance classification, where class data is unbalanced. For example, in a data set where the WR is 20% and there are an equal number of negative and positive bags, predicting only negative instances would achieve an accuracy of 90%. This is why the area under the receiver operating characteristic curve (AUC) and the area under the precision-recall curve (AUPRC) will be used in this paper as primary comparison metrics. To measure the ability of the algorithm to select a decision threshold, the F_1-scores will also be reported. The F_1-score is the harmonic mean between precision and recall. Since the negative bags are assumed to contain only negative instances,

they are not relevant for the comparison on witness identification, and thus, are ignored when measuring performance. For data sets generated several times, both the average results and standard deviations are reported.

Some algorithms have parameters that need to be optimized on the data. This is done via grid-search using 5-fold cross-validation on the entire data set. Since the instance labels are unknown, the performance of each configuration is evaluated using bag-level AUC. For the RSWI algorithm, two parameters were optimized. The dimensionality of the random subspaces ranged from 20% to 50% of the complete feature space dimensionality. The number of clusters used in k-means ranges from 30 to 120 with steps of 30. In all experiments, 2000 random subspaces were generated, this has proved to provide stable results in previous experiments. Fewer subspaces can be used especially with low-dimensional data sets, but since the method is computationally inexpensive, this parameter was not optimized. For all methods involving SVM, the regularization parameter (C) ranged from 0.1 to 10000 and the spread of the RBF kernel (γ), from 0.01 to 1000.

5.1 Reference Methods

SI-SVM: SI-SVM is an SVM trained using the labels assigned to bags as instance labels. It gives an indication on the pertinence of using MIL methods instead of regular supervised algorithms in a problem. The LIBSVM (Chang & Lin, 2011) implementation has been used.

CkNN-ROI: This method was selected because it was proposed for the identification of regions of interest (i.e. witness) in CIBR tasks. The method was implemented based on the details provided in the paper and the CkNN implementation provided on Zhou's website. The number of citers and references, ranging from 1 to 9 are chosen by grid-search cross-validation.

MI-SVM & mi-SVM: The two algorithms were implemented as described in the original paper (Andrews *et al.*, 2002). The LIBSVM (Chang & Lin, 2011) implementation has been used, and the parameters were optimized at each algorithm iteration.

EM-DD: The method has been selected as a reference method because it is the algorithm with the closest objective to the proposed method. The implementation provided with the MIL toolbox was used (Tax & Cheplygina, 2015). The algorithm was reinitialized 20 times, starting at the position of a random instance belonging to a positive bag. Only the result from the best run is used.

MILES: This method has been selected because it performs well on benchmark data sets and because the authors provided a way to use their algorithm for instance naming. The implementation provided with the MIL toolbox (Tax & Cheplygina, 2015) has been used.

KI-SVM: This method has been selected because it has been designed to find the key instance (i.e. witness) in bags. Since the bag-level version is a simplification of the instance-level version, only the instance-level version was used in this paper. The implementation provided by the authors on Zhou's website was used in the experiments.

5.2 Data Sets

Most existing MIL data sets do not provide annotation of individual instances. Therefore the Letters and Mammograms MIL data sets described below have been created using real-world data from existing data sets to evaluate MIL algorithms on the witness identification task.

Letters: This data set is created using the Letter Recognition data set introduced in (Frey & Slate, 1991). It contains a total of 20k instances of the 26 letters in the English alphabet. Each letter is encoded by a 16-dimensional feature vector. The reader is referred to the original paper for more details. A MIL version of the data set is created by grouping letters in bags. This allows control over WR and the number of positive concepts, which in this context, correspond to the different letters. A first collection of data sets is created by varying the number of positive concepts from 1 to 10. Each time a data set is generated, random letters are designated to be positive concepts, and all others are assigned to negative concepts. All bags contain 10 instances, and positive bags contain 2 instances from randomly selected from the positive concept. A second collection of data sets is generated to assess the effects of WR. The posi-

tive class is composed of 3 randomly selected concepts. Each bag contains 10 instances, and the number of witnesses in positive bags is determined by the WR. All data sets contain 100 positive and 100 negative bags. For each configuration, 10 different data sets are generated.

Birds: The birds data set was introduced in (Briggs *et al.*, 2012). In this data set, each bag corresponds to a 10 seconds recording of bird songs from one or more species. The recording is temporally segmented, and each part corresponds to a particular bird, or to background noises. These segments are the instances, each of represented by 38 features. Details on the features are given in the original paper. There are 13 types of bird in the data set. If one specie at a time is considered as the positive class, 13 MIL problems can be generated from this data set. Due to space constraints, only the results for the species providing the least and the most number of witnesses were reported. The entire data set contains a total 10232 instances, of which 32 belong to the hermit thrush and 1280 to the Hammond's flycatcher. The difficulty for MIL is that the WR is low and is not constant across positive bags.

Mammograms: This data set is created from the images contained in mini-MIAS database of mammograms (Suckling *et al.*, 1994). The database contains images of healthy patients, as well as patients exhibiting 1 of the 6 classes of abnormalities. For each abnormality, an image patch is extracted using the location annotations provided with the data set. These patches are positive instances, and negative instances are patches of various sizes extracted from tissue regions not intersecting with abnormalities regions, or from tissue regions belonging to healthy patients. Each patient is represented by a bag containing 10 patches. Because negative patches are extracted randomly, 5 versions of the data set are generated. The data set contains a total of 326 subjects, among which there are 117 subjects presenting abnormalities. Features are extracted from each patch. Similarly to (Kandemir *et al.*, 2014a), the feature vector contains the mean and standard deviation and a normalized 12-bin frequency histogram of the pixel intensities contained in the patch. This representation is augmented with the mean local binary pattern (LBP) extracted from a 13×13 pixel grid, and with the mean of densely extracted SIFT descriptors. Finally, the 5 Haralick features used in (Mudigonda *et al.*, 2000) are also used.

The resulting 220-dimensional vectors are reduced to 100-dimensional vectors using PCA. The difficulty for MIL is that the WR is low and there are 6 concepts in the positive class.

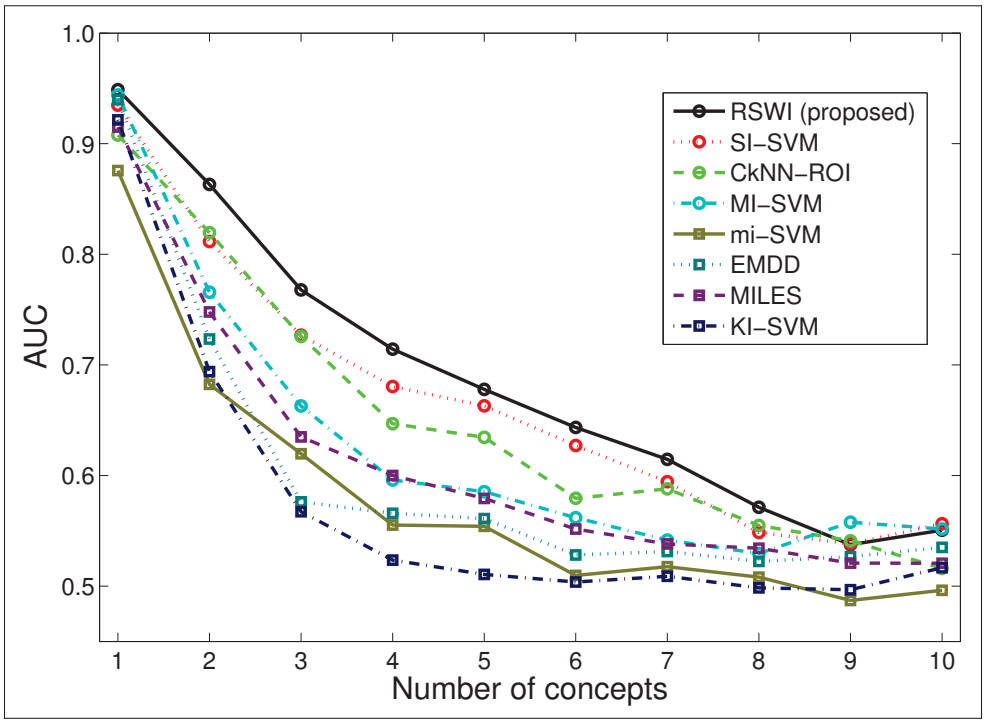


Figure-A I-2 Performance of MIL algorithms on the Letters data set depending on the number of positive concepts

6. Results

Figure I-3 and I-2 show the mean AUC of proposed and reference methods vs. the number of positive concepts and WR on the Letters data set. The AUPRC and F_1-score were not reported due to space constraints, and because they did not provide contrasting information to the AUC curve. In the number of concepts experiments, the performance of all algorithms decreases as the problem complexity increases. However, three methods, RSWI, CkNN and SI-SVM, are affected to a lesser extent. Both RSWI and CkNN-ROI are non-parametric methods, in which instances are classified based on bag distribution in their neighborhood. These local approaches provide robustness to distribution shape when compared to methods where an optimization process is performed using a global objective on all the data set. While CkNN-ROI is robust

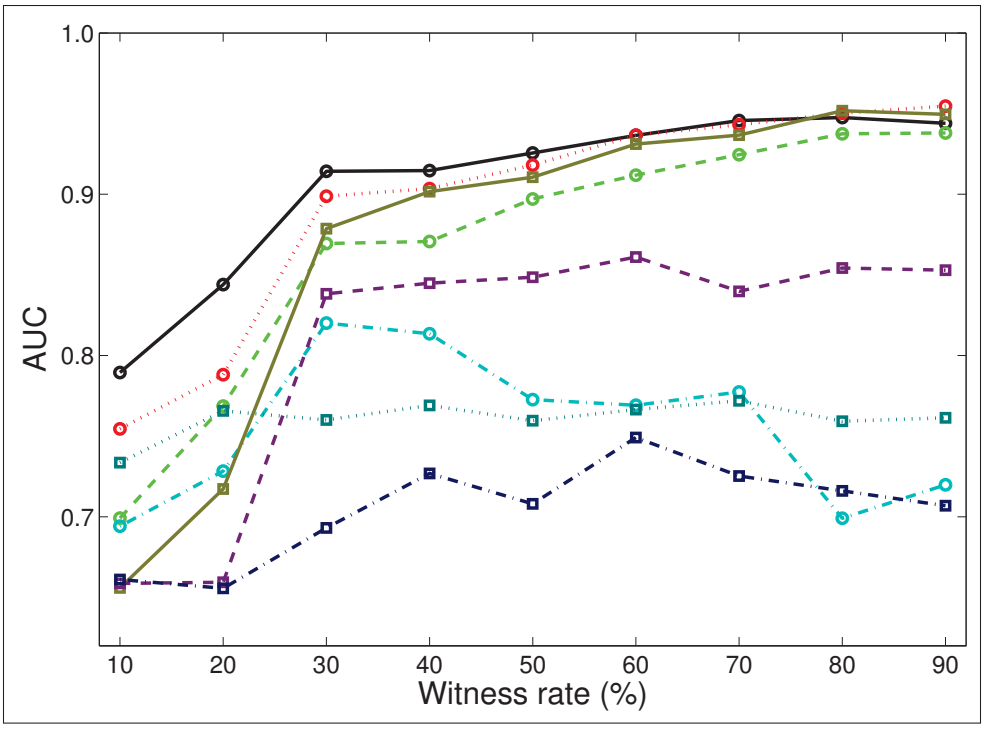


Figure-A I-3 Performance of MIL algorithms on the Letters data set depending on the witness rate

to the number of clusters, it is affected by low WR. Instances are labeled positive if they are close to any of the instances of a positive bag. If positive bags contain a large proportion of negative instances, it is more likely that negative test instances are found to be close to positive bags, which results in a high false positive rate. RSWI is affected by low WR to a lesser extent than all of the other methods. This is because witnesses are identified by comparing scores representing the proportion of instances from positive bags in their neighborhood. Even if this score is high in negative regions, it will still be lower than in positive regions of the feature space.

SI-SVM dominates all other SVM-based methods and EM-DD. It has been found in the past that in some application, SI-SVM may perform as well, and sometimes better than MIL algorithms (Ray & Craven, 2005). With SI-SVM, the problem is reduced to classification with a one-sided class noise. This reasonably applies to this task because the positive instances are organized in a small number of compact clusters, while negative instances are well distributed

Table-A I-1 Performance on the Mammograms and Birds MIL data set

Method	AUC (×100)	AUPRC (×100)	F_1-score (%)
Mammography (WR = 10%)			
SI-SVM	53.1 (5.8)	11.3 (2.3)	18.3 (0.2)
CkNN-ROI	56.7 (2.1)	14.6 (3.7)	20.2 (1.1)
MI-SVM	**69.3 (9.0)**	**26.6 (11.9)**	**26.4 (11.0)**
mi-SVM	53.4 (7.2)	13.7 (5.8)	18.8 (0.8)
EM-DD	55.6 (7.8)	13.6 (2.7)	8.8 (4.3)
MILES	65.5 (2.3)	24.0 (4.5)	23.8 (1.3)
KI-SVM	55.1 (10.1)	14.0 (6.3)	1.3 (2.1)
Proposed (RSWI)	67.4 (1.6)	26.2 (1.6)	24.1 (2.1)
Hermit Thrush (32/10232 witnesses)			
SI-SVM	61.1	12.4	8.6
CkNN-ROI	59.5	14.6	0.0
MI-SVM	59.2	16.4	5.2
mi-SVM	**70.7**	15.4	8.7
EM-DD	44.8	0.0	0.0
MILES	52.4	17.2	12.2
KI-SVM	37.1	7.3	0.0
Proposed (RSWI)	68.3	**20.5**	**29.1**
Hammond's Flycatcher (1280/10232 witnesses)			
SI-SVM	87.9	97.1	89.9
CkNN-ROI	89.4	97.6	89.6
MI-SVM	84.6	96.6	17.5
mi-SVM	89.0	97.6	**90.0**
EM-DD	89.2	97.8	58.9
MILES	74.8	93.7	55.0
KI-SVM	86.4	96.8	60.8
Proposed (RSWI)	**91.0**	**98.2**	86.6

in feature-space in a greater number of clusters. SI-SVM is the first iteration of mi-SVM. This indicates that the iterative optimization procedure of relabeling and training slowly converts positive regions of the feature space into negative regions. This happens when the number of positive instances is limited and distributed in many clusters. These positive regions become scanty, and thus, more susceptible to misclassification. However, as observed in Fig. I-3, when the WR increases mi-SVM performs comparably to SI-SVM.

As for KI-SVM and MI-SVM, during optimization, witnesses are selected under the constraint of bag classification accuracy. Only one instance per bag is selected, which is enough under

the standard MIL assumption to achieve high levels of bag classification accuracy. In a witness identification task, however, the goal is not to identify at least one witness, but all witnesses. If all bags contain positive instances from two or more concepts, the instances from one concept are predominantly selected, and thus, the others are ignored, which leads to poor performances. A similar argument can be made for MILES, which constructs a bag representation from an instance selection process governed by bag-level classification accuracy. EM-DD performance also declines when there is more than one concept. This is expected, since the algorithm searches for a single maximum of the DD function corresponding to the dominating concept. All other concepts are ignored.

The performance of the proposed and reference techniques on the Mammograms and Birds data sets is shown in Table I-1. The Mammograms data set has a low WR (10%) and is composed of multiple positive concepts corresponding to the 6 abnormality classes. MI-SVM is the best-performing algorithm despite the previous observation that this algorithm is affected by the presence of multiple concepts. In the Letters case, there is more than one witness per bag, although the algorithm selects only one during optimization. In the Mammograms data set, however, there is only one witness per bag, and thus selecting only one instance does not affect MI-SVM performance. The results obtained by RSWI are slightly lower to those obtained with MI-SVM. However, the results standard deviations indicate that RSWI achieves a high level of performance more consistently across all versions of the data set, which is a desirable property in practice.

The experiments on the Birds data set show the robustness of the proposed method to low WR. In the case of the Hermit Thrush, the witnesses represent only 0.3% of all instances. In such extreme conditions, many methods fail. For example, CkNN-ROI, EM-DD and KI-SVM cannot detect any of the witnesses, and thus, obtain a F_1-score of 0. SI-SVM and mi-SVM obtain appreciable results in terms of AUC but did not perform well in terms of F_1-score. Results suggest that both methods struggled to find an optimal classification threshold, which is the offset of the SVM hyper-plane. Both methods assume that all instances in positive bags are positive, which causes the SVM to include incorrectly labeled negatives in the positive

instance region. When the number of witnesses in the data set increases, as in the Hammond Flycatcher case, most algorithms perform comparably. However MI-SVM and KI-SVM do not achieve the performance level of their counterparts because both algorithms assume there is only one witness per bag which is not the case in this data set.

7. Conclusion

This paper presents a new MIL method for witness identification called RSWI. The proposed method achieves a high level of performance in all 3 tested applications, and demonstrated its applicability to problems with low WR and multiple positive concepts. The method is compared to 7 reference methods and obtains the best overall performance and consistently achieves first or second rank, while other methods perform unevenly across applications.

Future research will include methods to find a better classification threshold for the proposed and the reference methods. In addition, usability of RSWI as a component of a MIL algorithm should be explored.

ANNEX II

SCORE THRESHOLDING FOR ACCURATE INSTANCE CLASSIFICATION IN MULTIPLE INSTANCE LEARNING

Marc-André Carbonneau[1,2], Eric Granger[1], Ghyslain Gagnon[2]

[1] Laboratory for Imagery, Vision and Artificial Intelligence,
École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] Communications and Microelectronic Integration Laboratory,
École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article published in the proceedings of the 6th International Conference on Image Processing Theory Tools and Applications (IPTA), 2016.

Abstract

Multiple instance learning (MIL) is a form of weakly supervised learning for problems in which training instances are arranged into bags, and a label is provided for whole bags but not for individual instances. Most proposed MIL algorithms focus on bag classification, but more recently, the classification of individual instances has attracted the attention of the pattern recognition community. While these two tasks are similar, there are important differences in the consequences of instance misclassification. In this paper, the scoring function learned by MIL classifiers for the bag classification task is exploited for instance classification by adjusting the decision threshold. A new criterion for the threshold adjustment is proposed and validated using 7 reference MIL algorithms on 3 real-world data sets from different application domains. Experiments show considerable improvements in accuracy over these algorithms for instance classification. In some applications, the unweighted average recall increases by as much as 18%, while the $\mathbf{F_1}$-score increases by 12%.

2. Introduction

In multiple instance learning problems, instances are grouped into sets called bags. A label is provided for bags, but not for individual instances. The so-called standard MIL assumption (Amores, 2013) states that if a bag contains at least one positive instance, it is labeled as positive. Therefore, positive bags can contain a mixture of negative and positive instances, while negative bags contain only negative instances. Problems from many application domains can be formulated as MIL. In the past, it has been used for molecule activity prediction (Dietterich *et al.*, 1997), image classification (Chen *et al.*, 2006), computer-aided diagnosis (Kandemir & Hamprecht, 2015), visual object tracking (Babenko *et al.*, 2011c) and document classification (Zhou *et al.*, 2009). MIL research traditionally focused on bag classification, however, more recently, several authors considered problems in which instances must be classified individually (Zhou *et al.*, 2005b; Briggs *et al.*, 2012; Kandemir & Hamprecht, 2015; Carbonneau *et al.*, 2016c).

Typically, when MIL is applied to computer vision problems, images (or video) are divided in segments or patches. These segments correspond to instances, which are grouped in a bag representing the whole image. In this regard, MIL encompasses bag-of-words methods (Amores, 2013). For content-based image retrieval (CBIR) tasks, labels are assigned to bags and the exact label of the instances is not important. However, for image annotation tasks, such as object localization and tracking (Babenko *et al.*, 2011c), the instances must be classified individually (Cheplygina *et al.*, 2015d). This task is of significant importance, especially for computer-aided diagnosis, where regions of images are annotated as healthy or not. In this context, when using traditional supervised algorithms, the training data requires fine grained expert annotation which is costly (Kandemir & Hamprecht, 2015). With MIL, entire images can be used for learning and the patient diagnosis serves as weak supervision. This enables the use of an important quantity of training data otherwise unexploited.

It has been shown that the performance of MIL algorithms for bag classification is not representative of the performance for instance classification (Vanwinckelen *et al.*, 2015). This is

due to a combination of factors such as working assumptions on instance labels, the use of bag classification accuracy as optimization objective, and the data properties such as the witness rate (WR). Also, it can be shown experimentally that some algorithms perform well in terms of the area under the ROC curve (AUC) but provide low classification accuracy (Carbonneau *et al.*, 2016e). This suggests that some algorithms learn to score the instances correctly, but learn a suboptimal decision threshold to predict the instance or bag labels.

In this paper, the optimal decision threshold for bag classification is shown to be different from the optimal threshold for instance classification. Also, the threshold obtained by training MIL algorithms is experimentally shown to be suboptimal for the instance classification task. Finally, a criterion for the selection of the decision threshold is proposed to increase instance classification accuracy performance. The proposed criterion leverages the standard MIL assumption which states that instance labels in negative bags are fully known. The proposed criterion considers these instances individually, instead of in bags, which modifies the misclassification cost, and thus, raises accuracy at the instance level. The proposed criterion is used to adjust the decision threshold of 7 well-known reference MIL algorithms. Experiments are conducted on real-world data from 3 application domains.

The remainder of this paper is organized as follows: the next section surveys MIL algorithms applicable to instance classification problems. Section 4 shows how optimal thresholds for instance and bag classification are different, and introduces the proposed criterion for threshold adjustment. Finally, Section 5 presents the experimental methodology and the results are analyzed in Section 6.

3. Instance Classification in MIL

Several MIL methods originally proposed for bag classification, can be used directly for instance classification. These methods typically classify instances individually and then, under the standard MIL assumption, check for the presence of positive instances in bags. If a bag contains positive instances, it is labeled as positive, otherwise, it is labeled as negative. This

is the case for methods like APR (Dietterich *et al.*, 1997), MI-SVM and mi-SVM (Andrews *et al.*, 2002), RSIS (Carbonneau *et al.*, 2016e) and many diverse density (DD) based methods (Maron & Lozano-Pérez, 1998; Zhang & Goldman, 2001). When classifying bags with these methods, some types of instance classification error have no impact. For instance, in a positive bag, as long as at least one positive instance has been identified, false negatives and false positives have no effect on the bag label. This means that all but one positive instance per positive bag can be mislabeled, and yet, perfect bag accuracy can still be achieved. This is exploited directly by MI-SVM which selects only the most positive instance per positive bag to train the SVM. Other methods, like MILBoost (Babenko *et al.*, 2008) and EM-DD (Zhang & Goldman, 2001) use bag classification accuracy during their optimization process. This is a reasonable strategy for bag classification tasks but can be suboptimal for instance classification.

A large proportion of MIL methods do not attempt to classify all instances individually, but instead, consider entire bags as single objects. Some of these methods use kernels or set distance metrics to compare entire bags (Cheplygina *et al.*, 2015a; Gärtner *et al.*, 2002; Zhou *et al.*, 2009; Wang & Zucker, 2000), while other methods embed bags in a single vector representation (e.g. using distances to prototypes (Chen *et al.*, 2006)). Since these methods do not attempt to discover the label of individual instances, they generally cannot be applied to instance classification problems. There are, however, some bag-level methods that can be used for instance classification. For instance, MILES (Chen *et al.*, 2006) represents bags as sets of distances from selected instance prototypes. The authors proposed to use the contribution of each instance to the bag label as a witness identification mechanism. Other methods are adaptation of bag-level methods for instance classification. For instance, CkNN-ROI (Zhou *et al.*, 2005b) classifies bags using the minimal Hausdorff distance and the reference and citations scheme of CkNN (Wang & Zucker, 2000). Once a bag is deemed positive, each instance it contains is treated as a bag, and is classified individually. All of these methods were proposed to classify bags and thus, have consequent optimization objectives and working assumptions, which limits their accuracy for instance classification tasks.

4. Threshold for Instance Classification

This section describes why decision thresholds learned by MIL algorithms are often suboptimal for instance classification. Then, a new threshold selection criterion is proposed to increase the instance-level accuracy by making better use of the weak supervision available in MIL problems.

4.1 Decision Thresholds: Bags *vs*. Instances

Following the standard MIL assumption, the label of instances from negative bags are known without ambiguity while the labels of the instances in positive bags are unknown. Instance-based MIL methods infer the label of instances in order to predict bag labels. Generally speaking, to assign a hard label to an instance or a bag, a decision threshold is applied to a score. For several reasons described below, the optimal threshold for instance classification is often different than for bag classification.

Firstly, in many MIL problems, the proportion of positive instances in positive bags is low. For example, in images, most of the regions do not correspond to the object of interest and thus the positive bags exhibit low WR (Zhang *et al.*, 2002). This affects many MIL algorithms, like SI-SVM, EMDD, APR and CkNN, which assume that all instances in positive bags are positive. Also several MIL algorithms implicitly assume that the instances are independent and identically distributed (i.i.d.) in bags. However, this is rarely the case in practice. In many applications, there is some correlation between the positive and negative instances of the same bag (Zhou *et al.*, 2009). For example, in image classification, a tiger is most likely to be found in the jungle than in a spaceship. While instances corresponding to the jungle are as negative as instances from spaceships, the jungle instances are correlated with tiger instances. Moreover, the different segments of the same image share some similarities because of capture conditions. All the segments of an image with low illumination will be darker. In the drug activity prediction problem (Dietterich *et al.*, 1997), each bag contains many conformations of the same molecule. Only some of these conformations produce an effect of interest, but since

all instances represent the same molecule, they are likely to be similar to some extent. Finally, as stated in Section 3, several MIL algorithms, like MI-SVM and MIL-Boost, use the bag-level classification accuracy as an optimization criterion which is often suboptimal for instance classification.

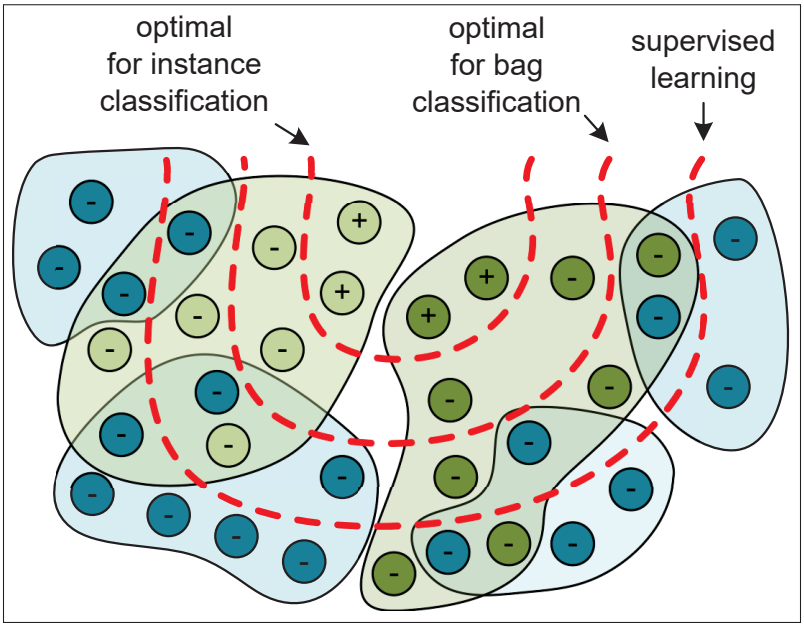


Figure-A II-1 Illustrative example of how different optimization objectives yield different threshold value in non i.i.d. instances and low WR MIL data sets. Positive bags are represented by green regions and the negative bags by blue regions

Fig. II-1 illustrates how low WR, correlation of instances in bags and optimization on bag-level accuracy can cause MIL algorithms to learn a suboptimal threshold for instance-level classification. In this example, positive bags are represented by green regions and the negative bags by blue regions. The instances in each bag are grouped together (correlated), and there is only a small number of positive instances in both positive bags. The dotted red lines are iso-contour of the score function learned by the classifier. In this illustrative example, there is a value for the decision threshold that can achieve a perfect classification of the instances, and thus, the bags. However, MIL algorithms optimizing bag-level accuracy can learn a different decision threshold, which also achieves a perfect bag classification. It only requires the exclusion of

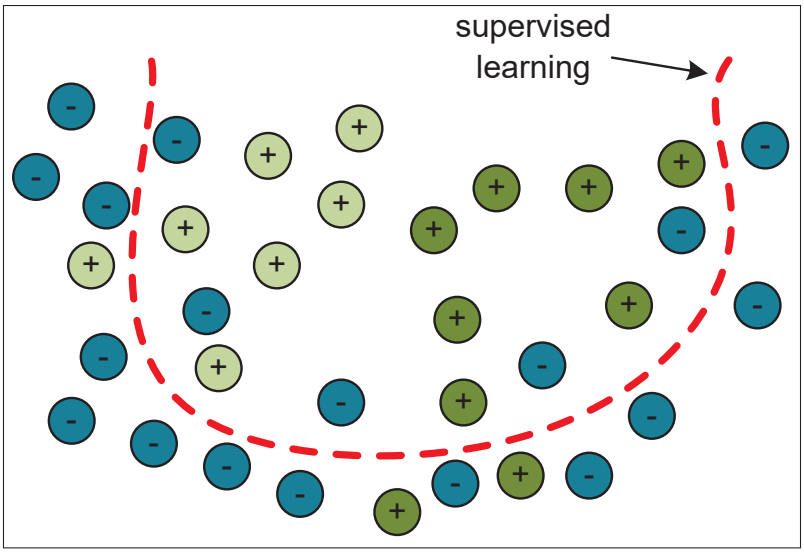


Figure-A II-2 The problem of the previous figure as seen by regular supervised algorithms: All instances inherited the label of their respective bag

all instances belonging to negative bags from the positive region. This produces false positives (FPs) in positive bags, which have no consequence when performing bag classification. However, in instance classification problems these FPs hinder performance. Finally, Fig. II-1 also shows a decision threshold that would learn a supervised algorithm like SI-SVM. In that case, all instances in positive bags are considered positive and the problem reverts to a regular supervised problem as illustrated in Fig. II-2. This shows why supervised algorithms are not suitable for instance classification in problems with low WR and non-i.i.d. instances.

4.2 Proposed Strategy for Threshold Adjustment

The proposed procedure aims at increasing the performance of existing MIL in the context of instance classification. The procedure is applied after an algorithm has undergone its usual process. The decision threshold is then updated to maximize the proposed criterion. Following the standard MIL assumption, two sources of information are reliable: the bag labels and the labels of the instances in negative bags. Both these sources are considered in the criterion, instead of using only bag labels like in most existing MIL methods.

Let $\mathcal{B}^+ = \{B_1, ..., B_{N^+}\}$ and $\mathcal{B}^- = \{B_1, ..., B_{N^-}\}$ be sets containing all positive and negative bags respectively. Each bag $B_i = \{\mathbf{x}_i^1, ..., \mathbf{x}_i^{M_i}\}$ is a set of instances. Finally, $\mathcal{I}^-$ is a set containing all instances of all negative bags $\mathcal{B}^-$. The threshold β is obtained by maximizing:

$$\beta = \underset{\beta}{\operatorname{argmax}} \{A_i(\beta) + A_b(\beta)\}. \tag{A II-1}$$

$A_i(\beta)$ is the instance level accuracy on all instances contained in negative bags:

$$A_i = \frac{TN^{\mathcal{I}^-}(\beta)}{|\mathcal{I}^-|} \tag{A II-2}$$

where $TN^{\mathcal{I}^-}$ is the number of correctly classified instances (true negatives). This diminishes the impact of misclassifying a single instance in a negative bag, which improves accuracy at instance-level. For example, if 1% of the instances in each negative bag are misclassified, then all these bags are misclassified, while 99% of the instance are correctly classified. The accuracy on the positive class must also be enforced. Since the instance labels in positive bags are unknown, the positive bag accuracy is used instead as the second term of the objective function:

$$A_b = \frac{TP^{\mathcal{B}^+}(\beta)}{|\mathcal{B}^+|} \tag{A II-3}$$

where $TP^{\mathcal{B}^+}$ is the number of correctly classified positive bags. By considering instances from negative bags individually the criterion reduces the penalty for FPs, which allows the identification of more positive instances. This results in an improved recall and ultimately an increased accuracy. In some applications, increasing recall is important: for example, in computer-aided diagnosis, a false negative could mean that a patient will not be diagnosed, and thus not treated.

5. Experimental Methodology

To measure the impact of the new threshold adjustment procedure on the performance of MIL algorithms, it has been applied to 7 reference algorithms, and on data sets from 3 application

domains. In MIL instance classification tasks, the classes are often imbalanced. Classification performance will therefore mainly be compared using two metrics that are appropriate for this context: the unweighted average recall (UAR), which is equivalent to averaging the accuracy for each class, and the F_1-score which is the harmonic mean of precision and recall. Precision, recall, the area under the precision-recall curve (AUC_{PR}) and the false positive rate (FPR) will also be reported to better understand the impact of the proposed threshold adjustment procedure for each class.

A bag-level stratified 10-fold cross-validation process was used to measure average performance. The hyper-parameters of all algorithms were optimized in each experiment via grid-search in a nested cross-validation. The adjustment of the decision threshold is performed on the training data.

5.1 Data Sets

This subsection describes the data sets used in the experiments. They are some of the few MIL benchmarks data sets providing ground truth for instance labels. They have been chosen because they each pose different types of challenges.

Birds (Briggs *et al.*, 2012): In this data set, each bag corresponds to a 10 second recording of bird songs from one or more species. The recording is temporally segmented, and each part corresponds to a particular bird, or to background noises. These 10232 segments are the instances, each represented by 38 features. Details on the extraction of these features are given in the original paper. There are 13 types of bird in the data set. If one species at a time is considered as the positive class, 13 MIL problems can be generated from this data set. The difficulty for MIL is that the WR is low and not constant across bags. Also there is sometimes a severe class imbalance at bag level.

Newsgroups (Settles *et al.*, 2008): This set was derived from the *20 Newsgroups* data set corpus. It contains posts from newsgroups on 20 subjects represented by 200 term frequency-inverse document frequency features. These features are generally sparse vectors, where each

element represents a word frequency in a text. When one of the subjects is selected as the positive class, all of the 19 other subjects are used as the negative class. The average WR of the data set is 3.7% which makes the problem difficult. Moreover, the distribution is highly multimodal.

SIVAL (Rahmani *et al.*, 2005): This benchmark data set is often used to compare MIL algorithms on image retrieval tasks. It contains 1500 images each segmented and manually labeled by (Settles *et al.*, 2008). There are 25 classes of complex objects photographed from different view-points in various environments. Each object is in turn considered as the positive class thus yielding 25 different learning problems. The bags correspond to images partitioned in approximately 30 segments, each corresponding to an instance. A segment is described by a 30-dimensional feature vector encoding color, texture and information about the neighboring segments. There are 60 images in each class, which makes 60 positive bags, and 5 images are randomly selected from each of the 24 other classes to create 120 negative bags. The WR of the data set is 25.5% in average but ranges from 3.1% to 90.6%. Moreover, the instances are non-i.i.d. as in many image data sets.

5.2 Reference Methods

This subsection describes the 7 reference methods used in the experiments. These methods were selected because they are well-known and represent a wide spectrum of MIL algorithms suitable for instance classification.

SI-SVM and SI-kNN: A simple approach for instance classification is to transpose MIL problems into supervised classification problems, and use regular classifiers such as SVM. Each instance inherits the label of its bag and a classifier is trained on all instances. While not a MIL method *per se*, this method has been used as a reference point in many MIL papers (Gärtner *et al.*, 2002; Ray & Craven, 2005) because it indicates the pertinence of using MIL methods instead of regular supervised algorithms in such problems. In this paper, SVM (SI-SVM) and nearest neighbor classifiers (SI-kNN) will be used in the experiments. These methods are in-

teresting in the context of this paper because they discard bag information and treat instances individually.

MI-SVM and mi-SVM (Andrews *et al.*, 2002): With mi-SVM, a label is assigned to each instance. An SVM is trained based on the instance label assignation. The instances are then reclassified using the newly obtain SVM. The resulting labels are then assigned to each instance and the SVM is retrained. This procedure is repeated until the labels are stable. The training procedure is similar for MI-SVM except that only the most positive instance of each positive bag is used for training. These two methods were selected because they are established MIL reference methods, they both use transductive learning and are different from each other in their optimization objective: mi-SVM focuses on instances while MI-SVM focuses on bags.

EM-DD (Zhang & Goldman, 2001): Diverse Density (DD) (Maron & Lozano-Pérez, 1998) is a measure of the probability that a given point in the input feature space belongs to the positive class. It depends on the proportion of instances from positive and negative bags in the neighborhood. The highest point of the DD function corresponds to the positive concept from which are generated the witnesses. Instances are classified based on their proximity to this point. In EM-DD (Zhang & Goldman, 2001), the Expectation-Maximization algorithm is used to locate the maximum of the DD function. This algorithm has been selected to represent DD-based methods because it is the most widely used as reference method. The implementation from (Tax & Cheplygina, 2015) is used in the experiments.

MIL-Boost (Babenko *et al.*, 2008): The MIL-Boost algorithm used in this paper is essentially the same as gradient boosting (Friedman, 2001) except that the loss function is computed on bag classification error. The instances are classified individually, and their labels are combined to obtain bag labels using a derivable approximation of the max function. This method has been selected because it was proposed to perform instance classification. The implementation from (Tax & Cheplygina, 2015) is used in the experiments.

CkNN-ROI (Zhou *et al.*, 2005b): CkNN (Wang & Zucker, 2000) is an adaptation of kNN to MIL problems. The distance between two bags is measured using the minimal Hausdorff

distance. Intuitively, it is the shortest distance between any of the instances contained in the two bags. In addition to using a distance measure for bags, the neighborhood is a combination of the r-nearest bags to the test bag, and the bags containing the test bag in their c-nearest bags. Each of the $r+c$ bags cast a vote on the label of the test bag, and the majority rule is applied. The algorithm was adapted in (Zhou *et al.*, 2005b) to perform classification of instances. Basically, it consists in classifying all bags using CkNN. Then, in positive bags, the instances are classified individually as if they were bags. CkNN was selected because it is a well-known non-parametric method, which has been adapted for instance classification. The implementation of (Tax & Cheplygina, 2015) was used in the experiments.

6. Results

6.1 Decision Thresholds Instance and Bag Classification

Table-A II-1 Differences in performance of MIL methods following the application of the proposed threshold adjustment method

		Bag Level						**Instance Level**				
Method	**Dataset**	UAR (%)	Prec. (%)	Rec. (%)	FPR (%)	F_1 (%)	AUC$_{PR}$ ($\times$100)	UAR (%)	Prec. (%)	Rec. (%)	FPR (%)	F_1 (%)
CkNN-ROI	Birds	-4.2	-20.2	**16.4**	5.6	-4.5	-5.5	-6.7	-16.4	**9.3**	1.8	-5.5
	SIVAL	**0.2**	**1.8**	**6.8**	**-1.4**	**2.0**	-3.8	-1.1	**7.6**	**4.8**	**-7.2**	-0.3
EM-DD	Newsgroups	-2.6	-8.2	**58.8**	2.2	**25.7**	**13.9**	**1.9**	-19.3	**10.9**	13.3	-4.2
	Birds	**4.0**	-35.5	**27.8**	21.6	-3.9	**14.2**	**9.6**	-34.7	**26.7**	20.8	**5.6**
	SIVAL	-1.7	-25.7	**35.2**	25.7	**1.9**	**32.6**	**6.9**	-25.6	**16.2**	25.6	**13.3**
mi-SVM	Newsgroups	-5.2	-15.2	**16.4**	11.7	**2.8**	-4.3	**14.5**	-21.9	**9.4**	18.4	**0.2**
	Birds	-5.4	-19.9	**13.6**	1.5	-3.8	-7.1	-2.1	-23.0	**10.0**	4.6	-6.9
	SIVAL	-6.2	-3.1	-1.7	3.1	-4.1	-0.1	-2.7	-11.6	**3.5**	11.6	-11.2
MI-SVM	Newsgroups	-4.6	-26.0	**40.7**	10.0	**15.9**	0.0	**17.4**	-42.0	**37.4**	26.0	**1.9**
	Birds	**1.8**	-39.7	**26.0**	29.0	-9.5	-6.2	**5.0**	-34.5	**18.5**	23.8	**2.8**
	SIVAL	-7.8	-28.7	**30.3**	24.7	-2.8	-0.7	**7.6**	-24.7	**20.9**	20.7	**12.0**
MILBoost	Birds	-2.4	-27.1	**23.3**	19.4	-2.9	-8.8	**5.5**	-40.7	**39.0**	6.9	**10.3**
	SIVAL	**0.6**	-23.5	**24.5**	22.7	**1.6**	-0.4	**7.6**	-20.5	**16.4**	19.3	**17.6**
SI-kNN	Birds	**2.7**	-13.7	**12.0**	3.7	-1.9	-0.5	**1.2**	-16.7	**1.4**	6.7	-3.7
	SIVAL	**6.3**	**4.5**	-0.6	**-4.5**	**4.3**	-2.7	-1.4	**10.1**	-12.1	**-10.1**	**4.2**
SI-SVM	Newsgroups	**1.0**	-13.4	**20.0**	**-4.1**	**12.7**	-5.7	**18.6**	-18.0	**10.7**	0.5	**12.6**
	Birds	**9.7**	**7.2**	**6.1**	**-20.2**	**16.9**	0.0	-3.7	-2.4	-12.3	**-10.7**	**5.1**
	SIVAL	**16.1**	**16.7**	-5.5	**-16.7**	**13.1**	0.0	-8.5	**15.3**	-26.0	**-15.3**	**0.8**

The two top graphs in Fig. II-3 show the accuracy performance at bag- and instances-level obtained with different threshold values with MI-SVM on the Brown Creeper data set from the Birds data set collection. There are two curves for each fold: a blue one obtained on the training data and a red curve obtained with test data. The similar shapes of the UAR curves obtained with the training and test data indicate that there is not a significant loss of generalization when using the training data to adjust the threshold instead of a held out validation fold.

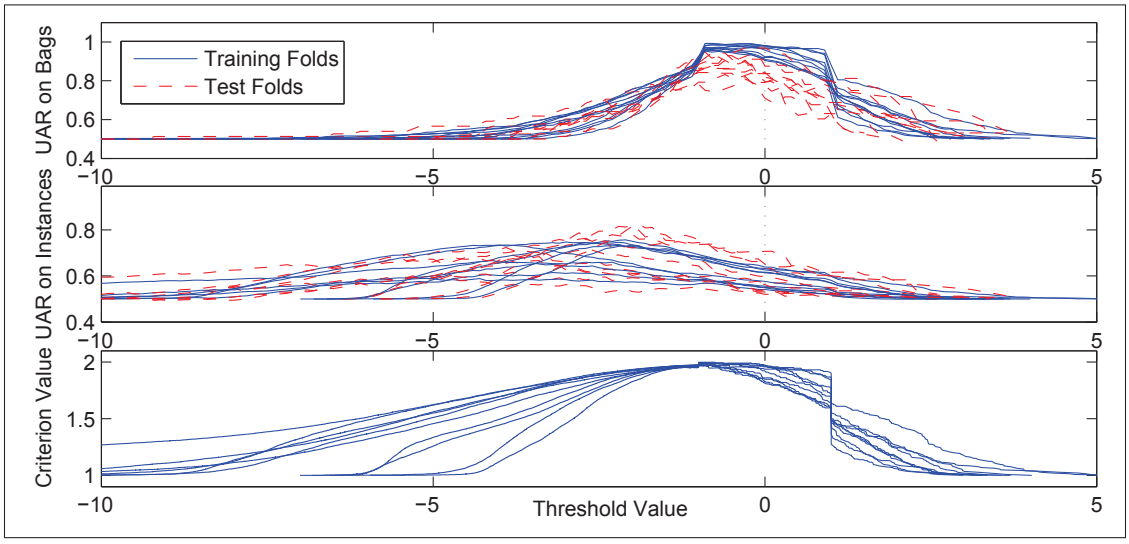


Figure-A II-3 Examples of classification accuracy obtained at different decision threshold values. Each line represents the UAR obtained using MI-SVM on a different fold on the Brown Creeper (Birds) data set. The blue lines are obtained with training folds, while the red dotted lines are obtained with test folds

When comparing these two graphs, it is clear that the optimal threshold for instance and bag classification are different. MI-SVM aims at classifying all instances from negative bags as negative and at least one instance per positive bag as positive. This indirectly optimizes bag-level accuracy, and as a result, the optimal threshold for bag-level classification is near 0, which is the threshold value used by an SVM. The graph suggests that using a threshold lower than 0 would improve accuracy at instance-level. As discussed in Section 4 the cost of misclassifying negative instances in negative bags, and positive instances in positive bags, are different in the two contexts which explains the different optimal threshold values.

The third curve, at the bottom, shows the value of the proposed criterion for the same threshold values as in the two other curves. While, the best threshold value according to the criterion is not optimal for instance classification, it represents an improvement on both performance measures in this case. The optimal threshold for instance classification cannot be learned because of the instance label uncertainty for instances belonging to positive bags.

6.2 Threshold Adjustment on Benchmark Data Sets

Table II-1 shows the difference on several performance metrics on the 3 corpus of data sets after applying the proposed threshold adjustment procedure[1] (e.g. UAR_{after} - UAR_{before}). The numbers are in bold when an improvement is obtained. The results for CkNN-ROI, SI-kNN and MILBoost are not reported for the Newsgroups data sets because these algorithms failed to learn and consistently yielded an UAR of 50.0% meaning that all bags were assigned the same label.

Results show that considerable improvement on instance classification performance can be obtained with the proposed criterion. For instance, on the Newsgroups data set, SI-SVM raises its UAR by 18.6% on average, or MILBoost increases its F_1-score by 17.6% on SIVAL. However, the table also indicates that the proposed method does not always lead to an improvement, and should not be applied blindly to all methods.

The adjustment strategy often lowers the decision threshold initially learned by the MIL algorithms. In other words, it makes the algorithm more sensitive to positive instances. As a result, after adjustment, recall is generally higher both for bag and instance classification, but precision is lower. Classes are highly imbalanced in MIL instance classification problems. For instance, in the Newsgroups data sets, the class imbalance ratio is 1:1 for bags but is 1:65 for instances. In that case, given perfect recall, if precision decreases by 50%, instance accuracy decreases by less than 1%. Thus, in this context, diminishing precision can still result in an improved instance-level accuracy. In many cases, the accuracy gain at instance-level does not

[1] The results on all individual data sets can be found on the author website: https://sites.google.com/site/marcandrecarbonneau/

reflect on bag-level accuracy. A more sensitive algorithm will be more susceptible to false positives, which have a different impact when classifying instances or bags.

The proposed method is particularly successful with methods using bag-level accuracy as an optimization criterion during learning. MI-SVM and MILBoost consistently improve their F_1-score and UAR for instance classification on all data sets. Similar results are observed for EM-DD, along with significant improvements on bag accuracy. The difference in maximizing the bag-level accuracy and the proposed criterion is that in the proposed criterion, bag accuracy is only measured on positive bags instead of on both classes. When computing bag accuracy on negative bags, a false positive has a great impact since it causes the entire bag to be misclassified. To correctly classify a positive bag, only one positive instance has to be identified. These two facts explain why algorithms maximizing bag accuracy are less sensitive. The proposed criterion lessens the penalty imposed to misclassified negative instances in negative bags by considering them individually instead of in groups.

Improvements were not consistently observed for all methods. The instance-level accuracy of the supervised methods, SI-SVM and SI-kNN, did not increase on the SIVAL data set. However UAR increased by 18.6% on the Newsgroups data set with SI-SVM, which suggests that the nature of the data distribution plays an important role in determining the success of the proposed method. In each experiment, the bag-level accuracy benefited from the threshold adjustment because these algorithms completely discard the structure of the MIL problem before learning. Therefore, they only optimize instance-level accuracy during learning. The proposed criterion also enforces accuracy at bag-level, which explains the accuracy improvement at this level. In essence, mi-SVM is similar to SI-SVM because the algorithm also classifies each instance individually. As a matter a fact, SI-SVM is the first iteration of the mi-SVM algorithm. Bag structure is only used if a positive bag does not contain a positive instance. In that case, the most positive instance is labeled as positive. This explains why mi-SVM behave similarly to SI-SVM. Finally, the proposed adjustment strategy did not prove beneficial to the CkNN-ROI algorithm on any data sets, perhaps because the algorithm makes predictions in two steps. It

starts by classifying bags and then, classifies instances. The proposed method is not equipped to deal with this kind of hierarchical decision process.

7. Conclusion

Instance and bag classification in MIL are different tasks that entail different objectives. It was shown that algorithms designed for bag classification can be used for instance classification. In that case, higher classification accuracy is achievable by adjusting the decision threshold. A criterion for threshold adjustment, which factors in bag labels and instance labels in negative bags, has been proposed. Experiments showed accuracy performance improvement for many bag classification methods used in instance classification tasks.

For future work, different criteria considering the cluster arrangement of the instance in feature space could be proposed for threshold adjustment. Also, research should be devoted to new methods incorporating instance classification criteria for the learning phase of the MIL algorithms instead of adjusting the threshold as a post-processing step. Finally, experiments should be conducted using larger data sets for which the criterion could be computed on a held-out validation set.

ANNEX III

REAL-TIME VISUAL PLAY-BREAK DETECTION IN SPORT EVENTS USING A CONTEXT DESCRIPTOR

Marc-André Carbonneau[1,2], Alexandre J. Raymond [1], Eric Granger[2], Ghyslain Gagnon[1]

[1] Communications and Microelectronic Integration Laboratory,
École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2]Laboratory for Imagery, Vision and Artificial Intelligence,
École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article published in the proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), 2015

Abstract

The detection of play and break segments in team sports is an essential step towards the automation of live game capture and broadcast. This paper presents a two-stage hierarchical method for play-break detection in non-edited video feeds of sport events. Unlike most existing methods, this algorithm performs action and event recognition on content, and thus does not rely on production cues of broadcast feeds. Moreover, the method does not require player tracking, can be used in real-time, and can be easily adapted to different sports. In the first stage, bag-of-words event detectors are trained to recognize key events such as line changes, face-offs and preliminary play-breaks. In the second stage, the output of the detectors along with a novel feature based on spatio-temporal interest points are used to create a context descriptor for the final decision. Experiments demonstrate the efficiency of the proposed method on real hockey game footage, achieving 90% accuracy.

2. Introduction

Automatic video summarization is of great importance in a world producing an ever increasing quantity of visual data. Cisco Systems Inc. forecasts that, in 2018, a million minutes of video

content will be transferred over the Internet every second (Cis, white paper, Cisco Systems Inc., June 2014). Sport events attract large audiences and therefore form a significant video category. In sport events, some sequences are less pertinent and do not catch the interest of the viewer (e.g. time-outs). When live broadcasting such events, it would make sense to detect these less interesting sequences to adjust the compression rate of the broadcast feed, or replace them with advertisement or relevant information. Play and break detection can be performed to achieve that goal. Also, the detection of these events from a fixed reference camera is essential to perform automatic editing and summarization of sporting videos.

Several approaches have been proposed to address play-break and event detection in sporting events. However, to our knowledge, none of these may be applied to unedited footage from a fixed camera because they rely on production cues. For instance (Tjondronegoro & Chen, 2010) uses production cues such as replay and close-up sequences. In (Wang & Zhang, 2012), Wang and Zhang proposed a method to recognize shooting events in ice hockey. The brightness of the frames was used as a feature to distinguish between close-ups and global camera views. Ekin (Ekin & Tekalp, 2003) also used the type of point of views in the frame as features. Qian (Qian *et al.*, 2011) used overlaid text amongst other features. Assfalg (Assfalg *et al.*, 2003) presented a method to detect important moments in a soccer game. This method uses unedited video streams from a mobile camera. The position of the ball is inferred based on the camera motion, and the part of the field covered in the frame is used as a feature. Therefore, the cameraman has performed most of the visual tracking and pattern recognition manually, which does not apply to the fixed camera context.

Recent advances in action and event recognition have made it possible to process the content of the video directly instead of focusing on its broadcast editing style, as existing methods do. The method proposed in this paper employs state-of-the-art action recognition methods to detect play and break segments on-line in an unedited video feed captured by a fixed camera. Moreover, it can run in real-time, and does not need segmentation or tracking of the players. Finally, the proposed method does not rely on rules or expert knowledge as in (Assfalg *et al.*, 2003; Chen *et al.*, 2004; Ariki *et al.*, 2006). Therefore, it can be adapted to other sports.

The main contribution of this paper is the introduction of a new method for play-break segmentation. The method is based on the standard bag-of-words (BoW) (Dollar *et al.*, 2005; Wang *et al.*, 2009; Peng *et al.*, 2014) classification pipeline adapted to event detection. Additionally, a new context descriptor based on the output of selected event detectors as well as spatio-temporal interest points (STIP) detection number is introduced.

The proposed system is validated on a new dataset consisting of a complete hockey game.

3. Event Detectors

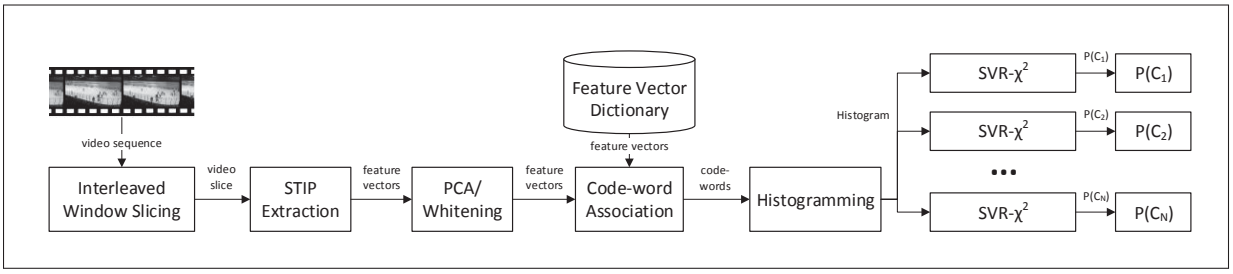


Figure-A III-1 Schematic block overview of the event detectors

The proposed method adapts the generic BoW classification framework to event detection. An overview of the detector stage is presented in Fig. III-1. This generic framework for action recognition (Dollar *et al.*, 2005; Wang *et al.*, 2009; Peng *et al.*, 2014) is used to classify complete sequences as one of some predefined classes. The proposed adapted framework enables the detection of events on a live feed even if they are concurrent.

First, the incoming frames are grouped in video slices. STIPs are then detected and extracted. Then, principal component analysis (PCA) and whitening are applied to the STIP feature vectors. Finally, histograms are produced and detection is performed on the slices. The rest of this section provides additional details on the event detectors.

3.1 Video Slicing

To achieve temporal localization of the events, the video sequence is divided into smaller subsequences called slices, using an overlapping sliding window (see. Fig. III-2). Each of these slices is classified separately and a likelihood score is produced for each of them. The step size between each window slice determines the granularity of the detection as well as the latency of the system when used in live feed contexts.

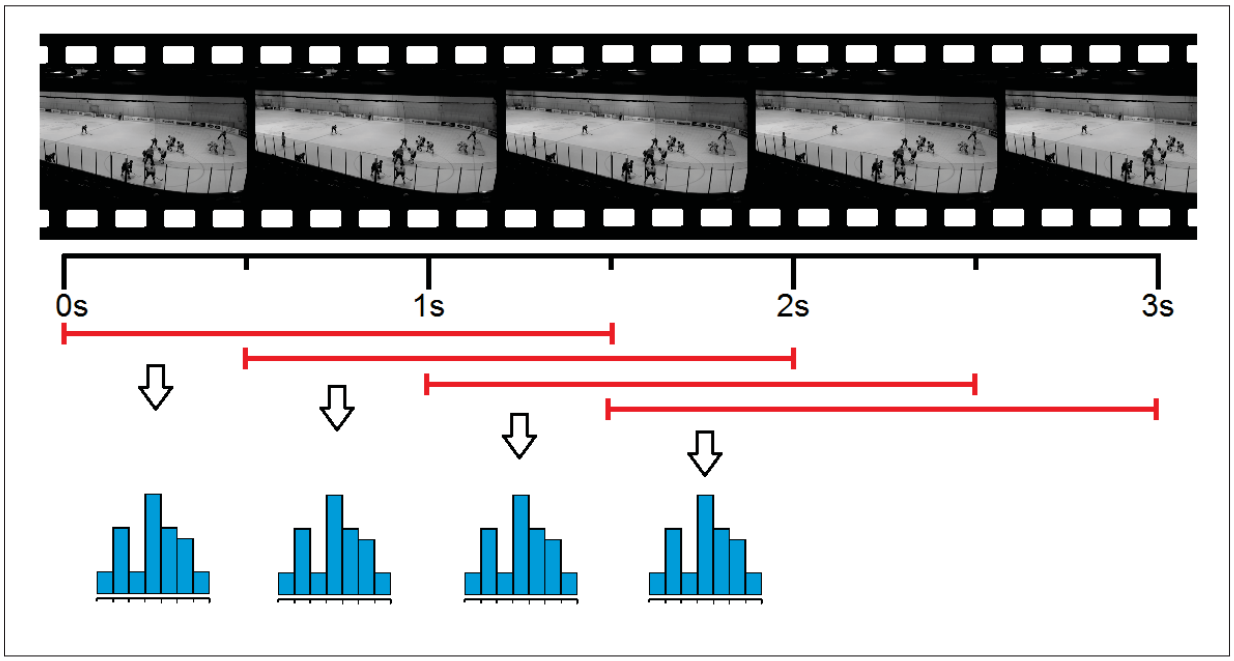


Figure-A III-2 Example of sliding window over a video sequence

3.2 Feature Extraction

To limit the amount of data to be processed, STIPs are detected and extracted. The detection of these points is achieved using a 3D adaptation of Harris corners introduced by Laptev in (Laptev, 2005). Each STIP is characterized by a combination of histograms of oriented gradients (HOG) and optical flow (HOF). This descriptor has been shown to be a reliable choice for action recognition (Wang *et al.*, 2009) because it can represent shape and motion. The

STIPs are detected and extracted at different scales to compensate for perspective effects in the images captured by a far-field camera. Wang's implementation of Laptev's algorithm (Wang *et al.*, 2009) was used in the following experiments.

Whitening and PCA projection and dimensionality reduction are applied to feature vectors to improve classification performance, as suggested in (Peng *et al.*, 2014).

3.3 Code-Word Association

In order to create a code-word dictionary, STIPs are randomly sampled from the complete training sequences. Samples taken from sequences containing the events to be recognized are also added to ensure these events are appropriately represented. If the STIPs were only sampled uniformly in the video sequences, there would be a risk of creating a dictionary lacking examples from rare classes. Once samples are collected, *k*-means clustering is performed in order to create *N* code-word prototypes. At runtime, every STIP feature vector is quantized to the nearest of these *N* prototypes using the Euclidean norm.

3.4 Detection

For each video slice, the code-words associated with the detected STIPs are pooled in a frequency histogram. This histogram represents the content of the slice. For every event detected, a likelihood score is obtained using the output of a support vector regression. The exponential χ^2 and the normalized χ^2 kernels are used (Chapelle *et al.*, 1999).

4. Context Descriptor

In order to improve the performance of the play-break recognition, a context descriptor, shown in Fig. III-3, is proposed. This descriptor is constructed using the likelihood scores from the event detectors described in Section 3. In the following experiments, three detectors have been trained to recognize *face-off*, *line change* and *play* sequences. These events contain informative cues regarding what is happening in the game and tend to precede or to indicate a play or a

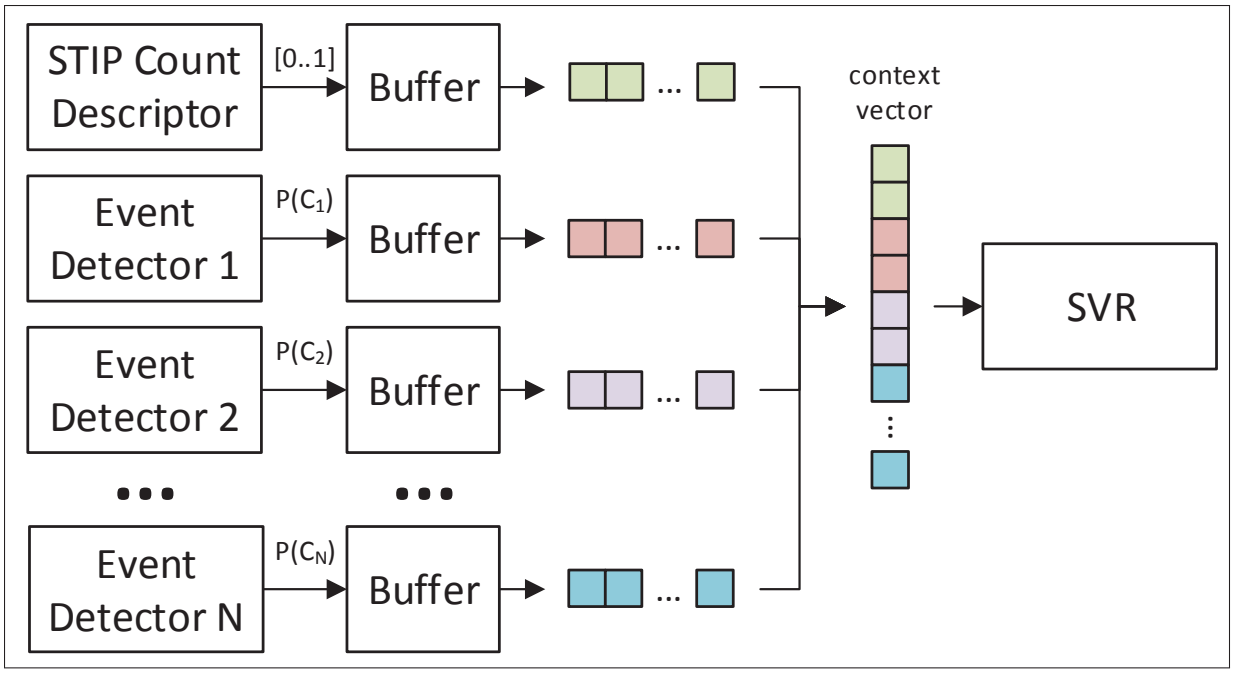


Figure-A III-3 Context vector construction

break sequence. For instance, knowing that a *face-off* just finished, there are greater chances that the present segment is a *play* segment. Also, if a long *line change* event is occurring, the game is probably in a *break*.

At time t, the context descriptor is given by:

$$\mathbf{c}_t = \{\mathbf{f}_t, \mathbf{l}_t, \mathbf{p}_t, \mathbf{s}_t\}, \tag{A III-1}$$

where the *face-off* descriptor $\mathbf{f}_t$ is given by:

$$\mathbf{f}_t = \{\theta_t, \theta_{t-1}, \ldots, \theta_{t-T}\}. \tag{A III-2}$$

T is the number of slices contained in the context window, and θ_t is the event detector output at time t. The *play* and *line change* descriptors $(\mathbf{l}_t, \mathbf{p}_t)$ are constructed in a similar manner. Along with the detector outputs, a descriptor $\mathbf{s}_t$, based on the number of STIPs in a slice, is also used. The elements of this vector are given by:

$$s = \begin{cases} M/\beta & \text{if } M < \beta; \\ 1 & \text{if } M \geq \beta, \end{cases} \tag{A III-3}$$

where M is the number of STIPs detected in a slice and β is a threshold that has to be set empirically. Each time a slice is produced, a new context vector is computed. The context vector is then classified as *play* or *break* using a support vector regression (SVR) with a radial basis function (RBF) kernel. A median filter is then applied considering the two last and two next predictions.

5. Experimental Methodology

5.1 Datasets

As no existing datasets met the requirements of our problem, a new one was created and made publicly available on-line[1]. The ÉTS dataset consists of a complete university-level hockey game captured from two far-field views of the ice rink. Fig. III-4 shows images taken from each camera. The video sequence from each point of view is processed as a different training instance, since their appearance differs considerably. The images are in grayscale with a 480x270 pixels resolution at 30 frame per second (fps). For our application, along with play-break classification, two other types of events are identified:

Play: A slice is labelled as a play slice when at least one player is visible and it is possible for a human to determine if the other players are actively playing.

Face-Off: A face-off event starts when the players are converging to their respective positions, waiting for the puck drop. It stops when the puck has been released and the players start to skate away. Some difficult examples include images taken from afar where only two or three players are visible.

[1] http://etsmtl.ca/Professeurs/ggagnon/Projects/ai-sports

Line Change: A line change event usually happens during a break, but may also occur during playtime. The dataset contains both situations. The event is characterized by players coming from and going to the player bench.

Figure-A III-4 Example of frames captured from each camera in the dataset

5.2 Protocol

A hockey game is divided in three periods. In the dataset, the game was captured from two angles, which yields 6 video sequences. The video sequences are further partitioned in six parts, making 36 sub-sequences. The experiment results are obtained using 6-fold cross-validation. For each fold, a part from each video sequence is reserved for testing. This is done to make sure every period and angle are represented equally in the testing and training set from both angles.

The sliding window contains 45 frames (1.5 seconds) and a new one is produced each time 15 frames are produced. The windows size has been chosen based on the results of previous experiments of the same type. In the ÉTS dataset this translates into 23,730 slices (14,312 play slices and 9,418 break slices). The dictionary contains 400 code-words. This has been arbitrarily selected based on earlier works on homogeneous datasets such as (Dollar *et al.*, 2005). The PCA stage is set to keep 97% of the signal energy, which corresponds to 83 to 85 components out of 162, depending on the sampled STIP used for the PCA computation.

The SVRs of the event detectors are trained on all positive examples from the training set. An equivalent number of negative examples are sampled randomly. The SVR regularization parameter C, kernel type and size γ are obtained by grid searching using 8-fold cross-validation on the training set. The configuration yielding the best accuracy is retained. The parameters for the context descriptor (T and β) and the final SVR (C and γ) classifier were determined using 8-fold cross-validation on the training set.

6. Results

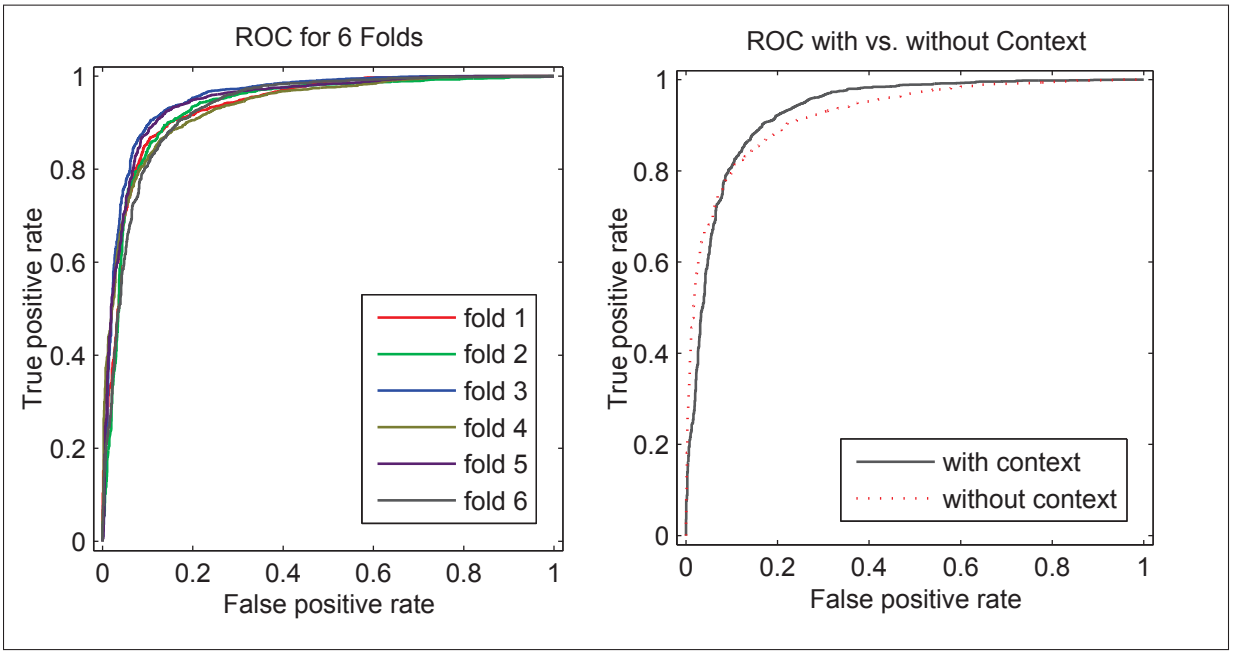


Figure-A III-5 ROC curves for play-break detection

Fig. III-5 shows an example of the receiver operating characteristic (ROC) curves obtained using the proposed methods. On the left side, the results for the 6 folds of an experiment run are presented when using context descriptors. The similarity of the curve indicates the stability of the proposed method's performance when testing with different data. The slight result variations from one fold to another are explained by the nature of the data. Some parts of the game contain fewer occurrences of *line change* and *face-off* events, which might affect

the training, and therefore the quality, of the event detectors. On the right side, the ROC curve for one fold is presented for the method with and without context descriptor.

To further assess the benefits of using the context descriptor stage, the results were averaged from 10 different runs of the 6-fold experiment on the entire dataset. Several replications of the experiment are needed since the recognition results depend on the dictionary quality which is affected by the randomly selected STIPs and k-means seeds. The average area under the ROC curve (AUC), equal error rate (EER) and accuracy are presented in Table III-1. The accuracy was measured when using the optimal threshold for the dataset. This optimal threshold corresponds to the intersection between the ROC curve and a diagonal starting from the upper left corner with a slope given by $-N/P$, where N is the number of break slices and P is the number of play slices in the dataset. Even without using the context descriptor, the proposed method achieves high accuracy (86.1%). However, an average performance boost of 3.87 $\pm$0.90% on the accuracy and 0.0102 $\pm$0.0071 on the AUC can be observed, which confirms the benefits of considering the temporal context in play-break classification. This represents a 28% reduction of the slice classification error.

Many misclassified video slices are situated at the start and end of a play event. Therefore, the proposed algorithm often disagree for 15 frames (500 ms) with the manually obtained labels. These slices, situated in this reasonable margin of error, represent 12.8% of the misclassified slices. This proportion rises to 22.6% when the acceptable margin is increased to 1000 ms. It should be noted that even for a human annotator, it is difficult to determine the exact duration of a play sequence, especially in the frequent situation where only one player is visible.

Table-A III-1 Average performance obtained using the proposed method

Algorithm	**AUC**	**EER** (%)	**Accuracy** (%)
without context	0.9322 $\pm$0.0055	14.25 $\pm$0.79	86.17 $\pm$0.75
with context	0.9424 $\pm$0.0072	11.95 $\pm$1.06	90.04 $\pm$0.99

The viability of the proposed method for real-time applications is assessed by measuring the processing time on a Intel Core i7 CPU. Using a single core, the algorithm detects STIPs, extracts the HOG/HOF features and saves them to a file at an average rate of 5 fps. Since image processing is highly parallelizable, one could expect to attain a frame rate greater than 30 fps using 8 cores. Once the STIPs are extracted, the analysis of a complete 20 minute period captured from 2 angles is performed in less than 150 seconds using a MATLAB implementation. In light of these results, meeting real-time requirements should be possible with current computer technology.

7. Conclusion

In this paper, we presented an efficient method for play-break detection. Unlike previous efforts in the field, our method does not require an edited video sequence or camera tracking of the action. Moreover, the method can be implemented in real-time, enabling its integration in automated capture systems. Experiments demonstrated the applicability of the algorithm to a real-life setting. The use of temporal context information proved to be beneficial to play-break segment recognition.

More experiments are needed in order to assess the suitability of this method to other camera angles, venues and sports. Also, the detection of other types of event should be explored to further increase its performance.

Acknowledgment

The work is supported by Quattriumm Inc. and the Natural Sciences and Engineering Research Council of Canada (NSERC). Thanks are also due to ReSMiQ for the partial support to this project.

BIBLIOGRAPHY

(white paper, Cisco Systems Inc., June 2014). *Cisco visual networking index : Forecast and methodology , 2013 – 2018*.

Addington, D. W. (1968). The relationship of selected vocal characteristics to personality perception. *Speech monographs*, 35(4), 492–503. doi: 10.1080/03637756809375599.

Aharon, M., Elad, M. & Bruckstein, A. (2006). K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *Ieee transactions signal processing*, 54(11), 4311–4322.

Alcala-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., Garcia, S., Sanchez, L. & Herrera, F. (2011). KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of multiple-valued logic and soft computing*, 17(2-3), 255–287.

Ali, K. & Saenko, K. (2014). Confidence-Rated Multiple Instance Boosting for Object Detection. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Alpaydın, E., Cheplygina, V., Loog, M. & Tax, D. M. (2015). Single- vs. multiple-instance classification. *Pattern recognition*, 48(9), 2831–2838.

Amores, J. (2010). Vocabulary-Based Approaches for Multiple-Instance Data: A Comparative Study. *Proceedings of the international conference on pattern recognition*.

Amores, J. (2013). Multiple Instance Classification: Review, Taxonomy and Comparative Study. *Artificial intelligence*, 201, 81–105.

Andrews, S., Tsochantaridis, I. & Hofmann, T. (2002). Support Vector Machines for Multiple-Instance Learning. *Proceedings of neural information processing systems*.

Argamon, S., Dhawle, S., Koppel, M. & Pennebaker, J. W. (2005). Lexical predictors of personality type. *Joint annual meeting of the interface and the classfication society of north america*.

Ariki, Y., Kubota, S. & Kumano, M. (2006). Automatic Production System of Soccer Sports Video by Digital Camera Work Based on Situation Recognition. *Multimedia, 2006. ism'06. eighth ieee int. symposium on*, pp. 851–860. doi: 10.1109/ISM.2006.37.

Assfalg, J., Bertini, M., Colombo, C., Bimbo, A. D. & Nunziati, W. (2003). Semantic Annotation of Soccer Videos: Automatic Highlights Identification. *Computer vision and image understanding*, 92(2-3), 285–305.

Auer, P. (1997). On Learning From Multi-Instance Examples: Empirical Evaluation of a Theoretical Approach. *Proceedings of the international conference on machine learning*.

Auer, P. & Ortner, R. (2004). A Boosting Approach to Multiple Instance Learning. *Proceedings of the 15th european conference on machine learning*, pp. 63–74.

Babenko, B. (2008). *Multiple Instance Learning : Algorithms and Applications*. San Diego, USA.

Babenko, B., Dollár, P., Tu, Z. & Belongie, S. (2008). Simultaneous Learning and Alignment: Multi-Instance and Multi-Pose Learning. *Proceedings of the europeen conference on computer vision*.

Babenko, B., Verma, N., Dollár, P. & Belongie, S. J. (2011a). Multiple Instance Learning with Manifold Bags. *Proceedings of the international conference on machine learning*.

Babenko, B., Yang, M.-H. & Belongie, S. (2011b). Robust Object Tracking with Online Multiple Instance Learning. *Ieee transactions pattern analysis machine intelligence*, 33(8), 1619–1632.

Babenko, B., Yang, M.-H. & Belongie, S. (2011c). Robust Object Tracking with Online Multiple Instance Learning. *Ieee transactions pattern analysis machine intelligence*, 33(8), 1619–1632.

Baldi, P., Cranmer, K., Faucett, T., Sadowski, P. & Whiteson, D. (2016). Parameterized machine learning for high-energy physics. *arxiv preprint arxiv:1601.07913*.

Bandyopadhyay, S., Ghosh, D., Mitra, R. & Zhao, Z. (2015). MBSTAR: multiple instance learning for predicting specific functional binding sites in microRNA targets. *Scientific reports*, 5, 8004.

Bao, H., Sakai, T., Sato, I. & Sugiyama, M. (2017). Risk Minimization Framework for Multiple Instance Learning from Positive and Unlabeled Bags. *arxiv preprint arxiv:1704.06767*.

Batliner, A., Steidl, S., Schuller, B., Seppi, D., Laskowski, K., Vogt, T., Devillers, L., Vidrascu, L., Amir, N., Kessous, L. & Others. (2006). Combining efforts for improving automatic classification of emotional user states. *Proceedings 5th slovenian and 1st international language technologies conference*.

Bengio, Y., Courville, A. & Vincent, P. (2013). Representation learning: A review and new perspectives. *Ieee transactions pattern analysis machine intelligence*, 35(8), 1798–1828.

Bergeron, C., Moore, G., Zaretzki, J., Breneman, C. M. & Bennett, K. P. (2012). Fast bundle algorithm for multiple-instance learning. *Ieee transactions pattern analysis machine intelligence*, 34(6), 1068-1079.

Bergeron, C., Zaretzki, J., Breneman, C. & Bennett, K. P. (2008). Multiple Instance Ranking. *Proceedings of the international conference on machine learning*.

Bergstra, J. & Bengio, Y. (2012). Random Search for Hyper-parameter Optimization. *Journal machine learning research*, 13(1), 281–305.

Bi, J. & Liang, J. (2007, jun). Multiple Instance Learning of Pulmonary Embolism Detection with Geodesic Distance along Vascular Structure. *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 1–8.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, USA: Springer.

Blum, A. & Kalai, A. (1998). A Note on Learning from Multiple-Instance Examples. *Machine learning*, 30(1), 23–29.

Boersma, P. & Weenink, D. (2001). Praat: doing phonetics by computer [computer program].

Bottou, L., Chapelle, O., DeCoste, D. & Weston, J. (2007). Support vector machine solvers. In *Large-Scale Kernel Machines* (pp. 1-27). MIT Press.

Branco, P., Torgo, L. & Ribeiro, R. P. (2016). A Survey of Predictive Modeling on Imbalanced Domains. *Acm computing surveys*, 49(2), 31:1—-31:50.

Breiman, L. (1996). Bagging Predictors. *Machine learning*, 24(2), 123–140.

Briggs, F., Fern, X. Z. & Raich, R. (2012). Rank-Loss Support Instance Machines for MIML Instance Annotation. *Proceedings of the acm international conference on knowledge discovery and data mining*.

Buisman, H. & Postma, E. (2012). BNAIC: The log-gabor method: Speech classification using spectrogram image analysis. *Proceedings of interspeech*.

Bunescu, R. & Mooney, R. (2007a). Learning to Extract Relations from the Web using Minimal Supervision. *Proceedings of the annual meeting of the association of computational linguistics*.

Bunescu, R. C. & Mooney, R. J. (2007b). Multiple Instance Learning for Sparse Positive Bags. *Proceedings of the international conference on machine learning*.

Cano, A., Zafra, A. & Ventura, S. (2015). Speeding up multiple instance learning classification rules on GPUs. *Knowledge and information systems*, 44(1), 127–145.

Carbonneau, M.-A., Raymond, A. J., Granger, E. & Gagnon, G. (2015, May). Real-time visual play-break detection in sport events using a context descriptor. *Proceedings of the ieee international symposium on circuits and systems*, pp. 2808–2811.

Carbonneau, M.-A., Cheplygina, V., Granger, E. & Gagnon, G. (2016a). Multiple Instance Learning: A Survey of Problem Characteristics and Applications. *Arxiv e-prints*, abs/1612.0.

Carbonneau, M.-A., Granger, E., Attabi, Y. & Gagnon, G. (2016b). Feature learning from spectrograms for assessment of personality traits. *arxiv preprint arxiv:1610.01223*.

Carbonneau, M.-A., Granger, E. & Gagnon, G. (2016c). Witness Identification in Multiple Instance Learning Using Random Subspaces. *Proceedings of the international conference on pattern recognition*.

Carbonneau, M.-A., Granger, E. & Gagnon, G. (2016d). Decision Threshold Adjustment Strategies for Increased Accuracy in Multiple Instance Learning. *Proceedings of the international conference on image processing theory, tools and application*.

Carbonneau, M.-A., Granger, E., Raymond, A. J. & Gagnon, G. (2016e). Robust multiple-instance learning ensembles using random subspace instance selection. *Pattern recognition*, 58, 83–99.

Chai, J., Chen, H., Huang, L. & Shang, F. (2014a). Maximum margin multiple-instance feature weighting. *Pattern recognition*, 47(6), 2091–2103.

Chai, J., Ding, X., Chen, H. & Li, T. (2014b). Multiple-instance discriminant analysis. *Pattern recognition*, 47(7), 2517–2531.

Chang, C.-C. & Lin, C.-J. (2011). LIBSVM: A Library for Support Vector Machines. *Acm transactions on intelligent systems and technology*, 2(3), 27:1—-27:27.

Chapelle, O., Haffner, P. & Vapnik, V. N. (1999). Support vector machines for histogram-based image classification. *Neural networks, ieee transactions on*, 10(5), 1055–1064.

Chastagnol, C. & Devillers, L. (2012). Personality Traits Detection Using a Parallelized Modified SFFS Algorithm. *Proceedings of interspeech*.

Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of artificial intelligence research*, 16(1), 321–357.

Chen, S.-C., Shyu, M.-L., Chen, M. & Zhang, C. (2004). A Decision Tree-Based Multimodal Data Mining Framework for Soccer Goal Detection. *Multimedia and expo, 2004. ieee international conference on*, 1, 265–268 Vol.1. doi: 10.1109/ICME.2004.1394176.

Chen, Y. & Wang, J. Z. (2004). Image Categorization by Learning and Reasoning with Regions. *Journal machine learning research*, 5, 913–939.

Chen, Y., Bi, J. & Wang, J. Z. (2006). MILES: Multiple-Instance Learning via Embedded Instance Selection. *Ieee transactions pattern analysis machine intelligence*, 28(12), 1931–1947.

Cheplygina, V., Tax, D. M. J. & Loog, M. (2015a). Dissimilarity-Based Ensembles for Multiple Instance Learning. *Ieee transactions on neural networks and learning systems*, 1–13.

Cheplygina, V. & Tax, D. M. J. (2015). Characterizing multiple instance datasets. *Proceedings of the international workshop on similarity-based pattern recognition*.

Cheplygina, V., Sørensen, L., Tax, D. M. J., Pedersen, J. H., Loog, M. & de Bruijne, M. (2014). Classification of COPD with multiple instance learning. *Proceedings of the international conference on pattern recognition*.

Cheplygina, V., Sørensen, L., Tax, D. M. J., Bruijne, M. & Loog, M. (2015b). Label Stability in Multiple Instance Learning. *Proceedings of medical image computing and computer assisted interventions conference*.

Cheplygina, V., Tax, D. M. & Loog, M. (2015c). Multiple instance learning with bag dissimilarities. *Pattern recognition*, 48(1), 264–275.

Cheplygina, V., Tax, D. M. & Loog, M. (2015d). On classification with bags, groups and sets. *Pattern recognition letters*, 59, 11–17.

Cinbis, R. G., Verbeek, J. & Schmid, C. (2016). Weakly Supervised Object Localization with Multi-fold Multiple Instance Learning. *Ieee transactions pattern analysis machine intelligence*.

Cohn, D. A., Ghahramani, Z. & Jordan, M. I. (1994). Active Learning with Statistical Models. *Proceedings of neural information processing systems*.

Cotton, C. V. & Ellis, D. P. W. (2011). Spectral vs. spectro-temporal features for acoustic event detection. *Proceedings of the ieee workshop on applications of signal processing to audio and acoustics*. doi: 10.1109/ASPAA.2011.6082331.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J. & Bray, C. (2004). Visual categorization with bags of keypoints. *Proceedings of the europeen conference on computer vision*.

Dasgupta, S. (2011). Two faces of active learning. *Theoretical computer science*, 412(19), 1767–1781.

Dasgupta, S. & Hsu, D. (2008). Hierarchical Sampling for Active Learning. *Proceedings of the international conference on machine learning*, pp. 208–215.

Daumé III, H. (2009). Frustratingly easy domain adaptation. *arxiv preprint arxiv:0907.1815*.

Demsar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal machine learning research*, 7, 1–30.

Deng, J., Zhang, Z., Marchi, E. & Schuller, B. (2013, Sep). Sparse autoencoder-based feature transfer learning for speech emotion recognition. *Proceedings of the international conference on affective computing and intelligent interaction*.

Dennis, J. W. (2014). *Sound Event Recognition in Unstructured Environments using Spectrogram Image Processing*. (Ph. D. thesis, Nanyang Technological University).

Dietterich, T. G., Lathrop, R. H. & Lozano-Pérez, T. (1997). Solving the Multiple Instance Problem with Axis-parallel Rectangles. *Artificial intelligence*, 89(1-2), 31–71.

Digman, J. M. (1996). The curious history of the five-factor model. *The five-factor model of personality*, 20.

Dollar, P., Rabaud, V., Cottrell, G. & Belongie, S. (2005). Behavior Recognition via Sparse Spatio-temporal Features. *Proceedings of the 14th international conference on computer communications and networks*, (ICCCN '05), 65–72.

Dooly, D. R., Zhang, Q., Goldman, S. A. & Amar, R. A. (2003). Multiple Instance Learning of Real Valued Data. *Journal machine learning research*, 3, 651–678.

Doran, G. (2015). *Multiple Instance Learning from Distributions*. (Ph. D. thesis, Case Western Reserve University).

Doran, G. & Ray, S. (2014a). A Theoretical and Empirical Analysis of Support Vector Machine Methods for Multiple-Instance Classification. *Machine learning*, 97(1-2), 79–102.

Doran, G. & Ray, S. (2014b). Learning Instance Concepts from Multiple-instance Data with Bags As Distributions. *Proceedings of the aaai conference on artificial intelligence*.

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004). Least angle regression. *The annals of statistics*, 32(2), 407–499.

Ekin, A. & Tekalp, M. (2003, Jul). Generic Play-break Event Detection for Summarization and Hierarchical Sports Video Analysis. *Multimedia and expo, 2003. proceedings international conference on*, 1, 169–72.

Eksi, R., Li, H.-D., Menon, R., Wen, Y., Omenn, G. S., Kretzler, M. & Guan, Y. (2013). Systematically differentiating functions for alternatively spliced isoforms through integrating RNA-seq data. *Plos computational biology*, 9(11).

EL-Manzalawy, Y., Dobbs, D. & Honavar, V. (2011). Predicting MHC-II Binding Affinity Using Multiple Instance Regression. *Ieee/acm transactions on computational biology and bioinformatics*, 8(4), 1067–1079.

Elad, M. & Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *Ieee transactions image processing*, 15(12), 3736-3745. doi: 10.1109/TIP.2006.881969.

Erdem, A. & Erdem, E. (2011). Multiple-Instance Learning with Instance Selection via Dominant Sets. *Proceedings of the international workshop on similarity-based pattern recognition*.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. (2010). The PASCAL visual object classes (VOC) challenge. *International journal of computer vision*, 88(2), 303–338.

Eyben, F., Scherer, K. R., Schuller, B. W., Sundberg, J., André, E., Busso, C., Devillers, L. Y., Epps, J., Laukka, P., Narayanan, S. S. & Truong, K. P. (2016). The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing. *Ieee transactions affective computing*, 7(2), 190–202.

Eyben, F., Weninger, F., Gross, F. & Schuller, B. (2013). Recent Developments in openSMILE, the Munich Open-source Multimedia Feature Extractor. *Proceedings of the acm conference on multimedia*.

Fang, H., Gupta, S., Iandola, F., Srivastava, R. K., Deng, L., Dollar, P., Gao, J., He, X., Mitchell, M., Platt, J. C., Lawrence Zitnick, C. & Zweig, G. (2015). From Captions to Visual Concepts and Back. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Foulds, J. & Frank, E. (2010). A Review of Multi-Instance Learning Assumptions. *The knowledge engineering review*, 25(1), 1–25.

Frank, E., Hall, M. A. & Witten, I. H. (2016). *The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques* (ed. Morgan Kaufmann).

Frenay, B. & Verleysen, M. (2014). Classification in the Presence of Label Noise: A Survey. *Ieee transactions neural networks learning systems*, 25(5), 845–869.

Freund, Y., Seung, H. S., Shamir, E. & Tishby, N. (1997). Selective Sampling Using the Query by Committee Algorithm. *Machine learning*, 28(2), 133–168.

Frey, P. W. & Slate, D. J. (1991). Letter recognition using holland-style adaptive classifiers. *Machine learning*, 6(2), 161–182.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The annals of statistics*, 29(5), 1189–1232.

Fu, Z. & Robles-Kelly, A. (2008, Dec). Fast multiple instance learning via L1,2 logistic regression. *Proceedings of the international conference on pattern recognition*, pp. 1–4.

Fu, Z., Robles-Kelly, A. & Zhou, J. (2011). MILIS: Multiple Instance Learning with Instance Selection. *Ieee transactions pattern analysis machine intelligence*, 33(5), 958–977.

Fuduli, A., Gaudioso, M. & Giallombardo, G. (2003). Minimizing nonconvex nonsmooth functions via cutting planes and proximity control. *Siam journal on optimization*, 14(3), 743–756.

Fujii, A., Tokunaga, T., Inui, K. & Tanaka, H. (1998). Selective Sampling for Example-based Word Sense Disambiguation. *Computational linguistics*, 24(4), 573–597.

Fung, G. M., Dundar, M., Krishnapuram, B. & Rao, R. B. (2007). Multiple Instance Learning for Computer Aided Diagnosis. In *Proceedings of the 21st Annual Conference on Neural Information Processing Systems Workshops (NIPS)*.

Garcia-Garcia, D. & Williamson, R. C. (2011). Degrees of supervision. *Proceedings of the conference on neural information processing systems workshops*, pp. 897–904.

Gärtner, T., Flach, P. A., Kowalczyk, A. & Smola, A. J. (2002). Multi-Instance Kernels. *Proceedings of the international conference on machine learning*.

Gehler, P. & Chapelle, O. (2007). Deterministic Annealing for Multiple-Instance Learning. *Aistats*.

Ghosh, S., Laksana, E., Morency, L. & Scherer, S. (2015). Learning representations of affect from speech. *arxiv preprint arxiv:1511.04747*.

Grauman, K. & Darrell, T. (2005). The pyramid match kernel: discriminative classification with sets of image features. *Proceedings of the international conference on computer vision*.

Grosse, R. B., Raina, R., Kwong, H. & Ng, A. Y. (2007). Shift-Invariance Sparse Coding for Audio Classification. *Proceedings of the conference on uncertainty in artificial intelligence*.

Gu, Y., Postma, E. & Lin, H.-X. (2015). Vocal Emotion Recognition with Log-Gabor Filters. *Proceedings of the audio-visual emotion challenge*. doi: 10.1145/2808196.2811635.

Guadagno, R. E., Okdie, B. M. & Eno, C. A. (2008). Who blogs? Personality predictors of blogging. *Computers in human behavior*, 24(5), 1993–2004. doi: 10.1016/j.chb.2007.09.001.

Guan, X., Raich, R. & Wong, W.-K. (2016). Efficient Multi-Instance Learning for Activity Recognition from Time Series Data Using an Auto-Regressive Hidden Markov Model. *Proceedings of the international conference on machine learning*.

Guillaumin, M., Verbeek, J. & Schmid, C. (2010). Multiple Instance Metric Learning from Automatically Labeled Bags of Faces. *Proceedings of the europeen conference on computer vision*.

Guo, Y. & Greiner, R. (2007). Optimistic Active Learning Using Mutual Information. *Proceedings of the international joint conference on artificial intelligence*.

Hamerly, G. & Elkan, C. (2004). Learning the k in k-means. In *Proceedings of Neural Information Processing Systems*.

Han, Y., Tao, Q. & Wang, J. (2010). Avoiding False Positive in Multi-Instance Learning. *Proceedings of neural information processing systems*.

Haralick, R., Shanmugam, K. & Dinstein, I. (1973). Textural features for image classification. *Ieee transactions on systems man and cybernetics*, SMC-3(6), 610-621. doi: 10.1109/TSMC.1973.4309314.

Hariharan, B., Arbeláez, P., Girshick, R. & Malik, J. (2014). Simultaneous Detection and Segmentation. *Proceedings of the europeen conference on computer vision*.

Harris, Z. S. (1954). Distributional structure. *Word*, 10, 146–162.

Hauptmann, A., Yan, R., Lin, W. H., Christel, M. & Wactlar, H. (2007). Can High-Level Concepts Fill the Semantic Gap in Video Retrieval? A Case Study With Broadcast News. *Ieee transactions on multimedia*, 9(5), 958–966.

Heckmann, M., Domont, X., Joublin, F. & Goerick, C. (2011). A hierarchical framework for spectro-temporal feature extraction. *Speech communications*, 53(5), 736–752. doi: 10.1016/j.specom.2010.08.006.

Herrera, F., Ventura, S., Bello, R., Cornelis, C., Zafra, A., Sánchez-Tarragó, D. & Vluymans, S. (2016a). *Multiple Instance Learning - Foundation and Algorithms*. Springer.

Herrera, F., Ventura, S., Bello, R., Cornelis, C., Zafra, A., Sánchez-Tarragó, D. & Vluymans, S. (2016b). Multiple Instance Multiple Label Learning. In *Multiple Instance Learning - Foundations and Algorithms* (ch. 9, pp. 191–206). Springer International Publishing.

Hoffman, J., Pathak, D., Darrell, T. & Saenko, K. (2015). Detector Discovery in the Wild: Joint Multiple Instance and Representation Learning. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Hoi, S. C. H., Jin, R. & Lyu, M. R. (2006). Large-scale Text Categorization by Batch Mode Active Learning. *Proceedings of the 15th international conference on world wide web*.

Hu, Y., Li, M. & Yu, N. (2008). Multiple-instance ranking: Learning to rank images for image retrieval. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Ikeuchi, K. (2014). *Computer Vision: A Reference Guide*. Springer.

Imam, T., Ting, K. M. & Kamruzzaman, J. (2006). z-SVM: An SVM for Improved Classification of Imbalanced Data. *Proceedings of the australian joint conference on artificial intelligence*.

Ivanov, V. & Chen, X. (2012). Modulation Spectrum Analysis for Speaker Personality Trait Recognition. *Proceedings of interspeech*.

J.-L. Shih, L.-H. C. (2002). Colour image retrieval based on primitives of colour moments. *Iee proceedings vision, image and signal processing*, 149, 370–376.

Jia, Y. & Zhang, C. (2008). Instance-level Semisupervised Multiple Instance Learning. *Proceedings of the aaai conference on artificial intelligence*.

Jorgensen, Z., Zhou, Y. & Inge, M. (2008). A Multiple Instance Learning Strategy for Combating Good Word Attacks on Spam Filters. *Journal machine learning research*, 9, 1115–1146.

Kandemir, M., Feuchtinger, A., Walch, A. & Hamprecht, F. A. (2014a). Digital pathology: Multiple instance learning can detect Barrett's cancer. *International symposium on biomedical imaging*, pp. 1348–1351.

Kandemir, M. & Hamprecht, F. A. (2015). Computer-aided diagnosis from weak supervision: a benchmarking study. *Computerized medical imaging and graphics*, 42, 44–50.

Kandemir, M., Zhang, C. & Hamprecht, F. A. (2014b). Empowering Multiple Instance Histopathology Cancer Diagnosis by Cell Graphs. *Proceedings of medical image computing and computer assisted interventions conference*.

Kang, F., Jin, R. & Sukthankar, R. (2006). Correlated label propagation with application to multi-label learning. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Karem, A. & Frigui, H. (2011). A multiple instance learning approach for landmine detection using Ground Penetrating Radar. *Proceedings of the ieee international geoscience and remote sensing symposium*.

Karpathy, A. & Fei-Fei, L. (2015). Deep Visual-Semantic Alignments for Generating Image Descriptions. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Keeler, J. D., Rumelhart, D. E. & Leow, W.-K. (1990). Integrated segmentation and recognition of hand-printed numerals. *Proceedings of neural information processing systems*.

Kim, S. & Choi, S. (2010). Local dimensionality reduction for multiple instance learning. *Proceeding of the ieee international workshop on machine learning for signal processing*.

Kim, Y., Lee, H. & Provost, E. M. (2013, May). Deep learning for robust feature generation in audiovisual emotion recognition. *Proceedings of the international conference on acoustics, speech, and signal processing*. doi: 10.1109/ICASSP.2013.6638346.

Kononenko, I. (1994). Estimating attributes: Analysis and extensions of RELIEF. *Proceedings of the european conference on machine learning*, pp. 171–182.

Konyushkova, K., Sznitman, R. & Fua, P. (2015). Introducing Geometry in Active Learning for Image Segmentation. *Proceedings of the international conference on computer vision*.

Kotzias, D., Denil, M., Blunsom, P. & de Freitas, N. (2014). Deep Multi-Instance Transfer Learning. *Corr*, abs/1411.3.

Kotzias, D., Denil, M., de Freitas, N. & Smyth, P. (2015). From Group to Individual Labels Using Deep Features. *Proceedings of the acm international conference on knowledge discovery and data mining*.

Kumar, A. & Raj, B. (2016). Weakly Supervised Scalable Audio Content Analysis. *Corr*, abs/1606.0.

Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*. Wiley.

Lai, K. T., Yu, F. X., Chen, M. S. & Chang, S. F. (2014). Video Event Detection by Inferring Temporal Instance Labels. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Lang, K. (1995). Newsweeder: Learning to filter netnews. *Proceedings of the international conference on machine learning*.

Laptev, I., Marszalek, M., Schmid, C. & Rozenfeld, B. (2008). Learning realistic human actions from movies. *Proceedins of the ieee conference on computer vision and pattern recognition*. doi: 10.1109/CVPR.2008.4587756.

Laptev, I. (2005). On Space-Time Interest Points. *International journal computer vision*, 64(2-3), 107–123. doi: 10.1007/s11263-005-1838-7.

Larochelle, H., Bengio, Y., Louradour, J. & Lamblin, P. (2009). Exploring Strategies for Training Deep Neural Networks. *Journal machine learning research*, 10, 1–40.

Lazebnik, S., Schmid, C. & Ponce, J. (2006). Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Lee, H., Battle, A., Raina, R. & Ng, A. Y. (2006). Efficient sparse coding algorithms. *Proceedings of neural information processing systems*.

Lee, H., Pham, P., Largman, Y. & Ng, A. Y. (2009). Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Proceedings of Neural Information Processing Systems*.

Leistner, C., Saffari, A. & Bischof, H. (2010). MIForests: Multiple-instance Learning with Randomized Trees. *Proceedings of the europeen conference on computer vision*.

Lewis, D. D. & Gale, W. A. (1994). A Sequential Algorithm for Training Text Classifiers. *Proceedings of the annual international acm sigir conference on research and development in information retrieval*.

Li, F. & Sminchisescu, C. (2010). Convex Multiple-Instance Learning by Estimating Likelihood Ratio. *Proceedings of neural information processing systems*.

Li, L.-j., Su, H., Fei-fei, L. & Xing, E. P. (2010). Object Bank: A High-Level Image Representation for Scene Classification & Semantic Feature Sparsification. In *Proceedings of Neural Information Processing Systems*.

Li, W. & Vasconcelos, N. (2015, Jun). Multiple instance learning for soft bags via top instances. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Li, W. J. & Yeung, D. Y. (2010). MILD: Multiple-Instance Learning via Disambiguation. *Ieee transactions on knowledge and data engineering*, 22(1), 76–89.

Li, Y., Tax, D. M., Duin, R. P. & Loog, M. (2013). Multiple-Instance Learning as a Classifier Combining Problem. *Pattern recognition*, 46(3), 865–874.

Li, Y.-F., Kwok, J. T., Tsang, I. W. & Zhou, Z.-H. (2009). A Convex Method for Locating Regions of Interest with Multi-instance Learning. *Proceedings of the joint european conference on machine learning and knowledge discovery in databases*.

Li, Z., Geng, G.-H., Feng, J., Peng, J.-y., Wen, C. & Liang, J.-l. (2014). Multiple instance learning based on positive instance selection and bag structure construction. *Pattern recognition letters*, 40, 19–26.

Lin, T.-Y., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C. L. (2014). Microsoft {COCO:} Common Objects in Context. *Corr*, abs/1405.0312.

Ling, C., Huang, J. & Zhang, H. (2003). AUC: A Better Measure than Accuracy in Comparing Learning Algorithms. In *Advances in Artificial Intelligence* (vol. 2671, pp. 329–341). Springer.

Loog, M. & Duin, R. P. W. (2012). The dipping phenomenon. *Structural, syntactic, and statistical pattern recognition: Joint iapr international workshop, sspr&spr*. doi: 10.1007/978-3-642-34166-3_34.

Loog, M., Krijthe, J. H. & Jensen, A. C. (2017). On measuring and quantifying performance: Error rates, surrogate loss, and an example in SSL. *Arxiv*, abs/1707.04025.

Lu, H., Zhou, Q., Wang, D. & Xiang, R. (2011). A co-training framework for visual tracking with multiple instance learning. *Proceedings of the ieee international conference and workshops on automatic face and gesture recognition*.

Lyon, R. F. (2010). Machine Hearing: An Emerging Field [Exploratory DSP]. *Signal processing magazine, ieee*, 27(5), 131–139.

Mairal, J., Bach, F., Ponce, J., Sapiro, G. & Zisserman, A. (2008). Discriminative learned dictionaries for local image analysis. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Mairal, J., Bach, F., Ponce, J. & Sapiro, G. (2009). Online Dictionary Learning for Sparse Coding. *Proceedings of the international conference on machine learning*. doi: 10.1145/1553374.1553463.

Mairesse, F., Walker, M. A., Mehl, M. R. & Moore, R. K. (2007). Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text. *Journal of artificial intelligence research*, 30(1), 457–500.

Mallat, S. (2008). *A wavelet tour of signal processing, third edition: The sparse way* (ed. 3rd). Academic Press.

Manandhar, A., Morton, K. D., Collins, L. M. & Torrione, P. A. (2012). Multiple instance learning for landmine detection using ground penetrating radar. *Proceedings of spie*.

Mandel, M. I. & Ellis, D. P. W. (2008). Multiple-instance learning for music information retrieval. *Proceedings of the 9th international conference of music information retrieval*.

Mangasarian, O. L. & Wild, E. W. (2008). Multiple Instance Classification via Successive Linear Programming. *Journal of optimization theory and applications*, 137(3), 555–568.

Mao, Q., Dong, M., Huang, Z. & Zhan, Y. (2014). Learning salient features for speech emotion recognition using convolutional neural networks. *Ieee transactions multimedia*, 16(8), 2203-2213. doi: 10.1109/TMM.2014.2360798.

Marĉelja, S. (1980). Mathematical description of the responses of simple cortical cells. *Journal of the optical society of america*, 70(11), 1297–1300.

Maron, O. & Lozano-Pérez, T. (1998). A Framework for Multiple-Instance Learning. *Proceedings of neural information processing systems*.

Maron, O. & Ratan, A. L. (1998). Multiple-Instance Learning for Natural Scene Classification. *Proceedings of the international conference on machine learning*.

Matsui, T., Goto, M., Vert, J.-P. & Uchiyama, Y. (2011). Gradient-based musical feature extraction based on scale-invariant feature transform. *Proceedings of the european signal processing conference*.

Mcgovern, A. & Jensen, D. (2003). Identifying Predictive Structures in Relational Data Using Multiple Instance Learning. *Proceedings of the international conference on machine learning*.

Meessen, J., Desurmont, X., Delaigle, J. F., Vleeschouwer, C. D. & Macq, B. (2007). Progressive Learning for Interactive Surveillance Scenes Retrieval. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Melendez, J., van Ginneken, B., Maduskar, P., Philipsen, R. H. H. M., Reither, K., Breuninger, M., Adetifa, I. M. O., Maane, R., Ayles, H. & Sánchez, C. I. (2015a). A novel multiple-instance learning-based approach to computer-aided detection of tuberculosis on chest x-rays. *Ieee transactions on medical imaging*, 34(1), 179-192.

Melendez, J., van Ginneken, B., Maduskar, P., Philipsen, R., Ayles, H. & Sanchez, C. (2015b). On combining multiple-instance learning and active learning for computer-aided detection of tuberculosis. *Ieee transactions on medical imaging*, PP.

Melendez, J., van Ginneken, B., Maduskar, P., Philipsen, R. H. H. M., Ayles, H. & Sánchez, C. I. (2016a). On Combining Multiple-Instance Learning and Active Learning for Computer-Aided Detection of Tuberculosis. *Ieee transactions medical imaging*, 35(4), 1013–1024.

Melendez, J., van Ginneken, B., Maduskar, P., Philipsen, R. H. H. M., Ayles, H. & Sánchez, C. I. (2016b). On combining multiple-instance learning and active learning for computer-aided detection of tuberculosis. *Ieee transactions on medical imaging*, 35(4), 1013-1024.

Melville, P. & Mooney, R. J. (2004). Diverse Ensembles for Active Learning. *Proceedings of the international conference on machine learning*.

Merler, M., Huang, B., Xie, L., Hua, G. & Natsev, A. (2012). Semantic Model Vectors for Complex Video Event Recognition. *Ieee transactions multimedia*, 14(1), 88–101.

Mohamed, A., Dahl, G. E. & Hinton, G. (2012). Acoustic modeling using deep belief networks. *Ieee transactions on audio, speech, and language processing*, 20(1), 14-22.

Mohammadi, G. & Vinciarelli, A. (2012). Automatic Personality Perception: Prediction of Trait Attribution Based on Prosodic Features. *Ieee transactions affective computing*, 3(3), 273–284.

Montacié, C. & Caraty, M.-j. (2012). Pitch and Intonation Contribution to Speakers ' Traits Classification. *Proceedings of interspeech*.

Morgan, N. (2012). Deep and wide: Multiple layers in automatic speech recognition. *Ieee transactions on audio, speech, and language processing*, 20(1), 7-13.

Mudigonda, N. R., Rangayyan, R. M. & Desautels, J. E. L. (2000). Gradient and texture analysis for the classification of mammographic masses. *Ieee transactions on medical imaging*, 19(10), 1032–1043.

Müller, A. & Behnke, S. (2012). Multi-instance Methods for Partially Supervised Image Segmentation. *Proceedings of the iapr international workshop on partially supervised learning*, pp. 110–119.

Muroi, T., Takashima, R., Takiguchi, T. & Ariki, Y. (2009). Gradient-based acoustic features for speech recognition. *Proceedings of the international symposium on intelligent signal processing and communication systems*. doi: 10.1109/ISPACS.2009.5383805.

Murray, J. F., Hughes, G. F. & Kreutz-Delgado, K. (2005). Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application. *Journal machine learning research*, 6, 783–816.

Nguyen, C.-T., Zhan, D.-C. & Zhou, Z.-H. (2013). Multi-modal Image Annotation with Multi-instance Multi-label LDA. *Proceedings of the international joint conference on artificial intelligence*.

Nguyen, H. T. & Smeulders, A. (2004). Active Learning Using Pre-clustering. *Proceedings of the international conference on machine learning*.

Nowak, E., Jurie, F. & Triggs, B. (2006). Sampling Strategies for Bag-of-Features Image Classification. *Proceedings of the europeen conference on computer vision*.

Palachanis, D. (2014). *Using the Multiple Instance Learning framework to address differential regulation*. (Master, Delft University of Technology).

Pappas, N. & Popescu-Belis, A. (2014). Explaining the Stars: Weighted Multiple-Instance Learning for Aspect-Based Sentiment Analysis. *Proceedings of the conference on empirical methods on natural language processing*.

Peng, X., Wang, L., Wang, X. & Qiao, Y. (2014). Bag of Visual Words and Fusion Methods for Action Recognition: Comprehensive Study and Good Practice. *arxiv preprint arxiv:1405.4506*.

Peyré, G. (2009). Sparse Modeling of Textures. *Journal of mathematical imaging and vision*, 34(1), 17–31. doi: 10.1007/s10851-008-0120-3.

Phan, S., Le, D.-D. & Satoh, S. (2015). Multimedia Event Detection Using Event-Driven Multiple Instance Learning. *Proceedings of the acm conference on multimedia*.

Philbin, J., Chum, O., Isard, M., Sivic, J. & Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. *Proceedings of the ieee conference on computer vision and pattern recognition*. doi: 10.1109/CVPR.2007.383172.

Ping, W., Xu, Y., Ren, K., Chi, C.-H. & Shen, F. (2010). Non-I.I.D. Multi-instance Dimensionality Reduction by Learning a Maximum Bag Margin Subspace. *Proceedings of the aaai conference on artificial intelligence*.

Ping, W., Xu, Y., Wang, J. & Hua, X.-S. (2011). FAMER: Making Multi-Instance Learning Better and Faster. *Proceedings of the siam international conference on data mining*.

Pohjalainen, J., Kadioglu, S. & Räsänen, O. (2012). Feature Selection for Speaker Traits. *Proceedings of interspeech*.

Provost, F. J., Fawcett, T. & Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. *Proceedings of the international conference on machine learning*.

Qi, G. J., Hua, X. S., Rui, Y., Mei, T., Tang, J. & Zhang, H. J. (2007). Concurrent multiple instance learning for image categorization. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Qian, X., Liu, G., Wang, H., Li, Z. & Wang, Z. (2011). Soccer Video Event Detection by Fusing Middle Level Visual Semantics of an Event Clip. In *Advances in Multimedia Information Processing* (vol. 6298, pp. 439–451). Springer Berlin Heidelberg. doi: 10.1007/978-3-642-15696-0_41.

Qiu, L., Lin, H., Ramsay, J. & Yang, F. (2012). You are what you tweet: Personality expression and perception on Twitter. *Journal of research in personality*, 46(6), 710–718.

Quellec, G., Lamard, M., Cozic, M., Coatrieux, G. & Cazuguel, G. (2016). Multiple-Instance Learning for Anomaly Detection in Digital Mammography. *Ieee transactions on medical imaging*, 35(7), 1604-1614.

Quellec, G., Cazuguel, G., Cochener, B. & Lamard, M. (2017). Multiple-instance learning for medical image and video analysis. *Ieee reviews in biomedical engineering*.

Quellec, G. et al. (2012). A multiple-instance learning framework for diabetic retinopathy screening. *Medical image analysis*, 16(6), 1228–1240.

Rahmani, R. & Goldman, S. A. (2006). MISSL: Multiple-instance Semi-supervised Learning. *Proceedings of the international conference on machine learning*.

Rahmani, R., Goldman, S. A., Zhang, H., Krettek, J. & Fritts, J. E. (2005). Localized Content Based Image Retrieval. *Proceedings of the acm sigmm international workshop on multimedia information retrieval*.

Raina, R., Battle, A., Lee, H., Packer, B. & Ng, A. Y. (2007). Self-taught learning: Transfer learning from unlabeled data. *Proceedings of the international conference on machine learning*.

Rammstedt, B. & John, O. P. (2007). Measuring personality in one minute or less: A 10-item short version of the Big Five Inventory in English and German. *Journal of research in personality*, 41(1), 203–212. doi: 10.1016/j.jrp.2006.02.001.

Ramon, Jan and De Raedt, L. (2000). Multi Instance Neural Networks. *Proceedings of the international conference on machine learning*.

Ray, S. & Craven, M. (2005). Supervised Versus Multiple Instance Learning: An Empirical Comparison. *Proceedings of the international conference on machine learning*.

Ray, S. & Page, D. (2001). Multiple Instance Regression. *Proceedings of the international conference on machine learning*.

Raykar, V. C., Krishnapuram, B., Bi, J., Dundar, M. & Rao, R. B. (2008). Bayesian Multiple Instance Learning: Automatic Feature Selection and Inductive Transfer. *Proceedings of the international conference on machine learning*.

Ren, W., Huang, K., Tao, D. & Tan, T. (2016). Weakly Supervised Large Scale Object Localization with Multiple Instance Learning and Bag Splitting. *Ieee transactions pattern analysis and machine intelligence*, 38(2), 405–416.

Ringeval, F., Sonderegger, A., Sauer, J. & Lalanne, D. (2013). Introducing the RECOLA multimodal corpus of remote collaborative and affective interactions. *Proceedings of the ieee international conference and workshops on automatic face and gesture recognition*.

Rosenberg, A. (2012). Classifying Skewed Data: Importance Weighting to Optimize Average Recall. *Proceedings of interspeech*.

Roy, N. & McCallum, A. (2001). Toward Optimal Active Learning Through Sampling Estimation of Error Reduction. *Proceedings of the international conference on machine learning*.

Rubner, Y., Tomasi, C. & Guibas, L. J. (2000). The Earth Mover's Distance As a Metric for Image Retrieval. *International journal computer vision*, 40(2), 99–121.

Ruffo, G. (2000). *Learning Single and Multiple Instance Decision Trees for Computer Security Applications*. (Ph. D. thesis, Department of Computer Science, University of Turin).

Ruiz-Muñoz, J. F., Orozco-Alzate, M. & Castellanos-Dominguez, G. (2015). Multiple Instance Learning-based Birdsong Classification Using Unsupervised Recording Segmentation. *Proceedings of the international joint conference on artificial intelligence*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International journal computer vision*, 115(3), 211–252.

Ryoo, M. S. & Aggarwal, J. K. (2009). Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. *Proceedings of the international conference on computer vision*.

Sabato, S. & Tishby, N. (2012). Multi-instance Learning with Any Hypothesis Class. *Journal machine learning research*, 13(1), 2999–3039.

Sadanand, S. & Corso, J. J. (2012). Action bank: A high-level representation of activity in video. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Sapienza, M., Cuzzolin, F. & Torr, P. H. S. (2014). Learning Discriminative Space–Time Action Parts from Weakly Labelled Videos. *International journal of computer vision*, 110(1), 30–47.

Schuller, B., Steidl, S., Batliner, A., Nöth, E., Vinciarelli, A., Burkhardt, F., Van Son, R., Weninger, F., Eyben, F., Bocklet, T., Mohammadi, G. & Weiss, B. (2012). The Interspeech 2012 Speaker Trait Challenge. *Proceedings of interspeech*.

Schuller, B., Steidl, S., Batliner, A., Hantke, S., Orozco-Arroyave, J. R., Nöth, E., Zhang, Y. & Weninger, F. (2015a). The Interspeech 2015 Computational Paralinguistics Challenge: Nativeness, Parkinson's & Eating Condition. *Proceedings of interspeech*.

Schuller, B., Steidl, S., Batliner, A., Nöth, E., Vinciarelli, A., Burkhardt, F., van Son, R., Weninger, F., Eyben, F., Bocklet, T., Mohammadi, G. & Weiss, B. (2015b). A Survey on perceived speaker traits: Personality, likability, pathology, and the first challenge. *Computer speech & language*, 29(1), 100–131. doi: 10.1016/j.csl.2014.08.003.

Schutte, K. T. (2009). *Parts-based Models and Local Features for Automatic Speech Recognition*. (Ph. D. thesis, Massachusetts Institute of Technology, Cambridge, USA).

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J. & Napolitano, A. (2010). RUSBoost: A Hybrid Approach to Alleviating Class Imbalance. *Ieee transactions on systems, man, and cybernetics*, 40(1), 185–197.

Settles, B. (2009). *Active Learning Literature Survey* (Report n°1648).

Settles, B. & Craven, M. (2008). An Analysis of Active Learning Strategies for Sequence Labeling Tasks. *Proceedings of the conference on empirical methods on natural language processing*.

Settles, B., Craven, M. & Ray, S. (2008). Multiple-Instance Active Learning. In *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems Workshops (NIPS)*.

Seung, H. S., Opper, M. & Sompolinsky, H. (1992). Query by Committee. *Proceedings of the annual workshop on computational learning theory*.

Shalev-Shwartz, S. & Srebro, N. (2008). Svm optimization: Inverse dependence on training set size. *Proceedings of the international conference on machine learning*.

Sharan, R. V. & Moir, T. J. (2015). Subband Time-Frequency Image Texture Features for Robust Audio Surveillance. *Ieee transactions information forensics security*, 10(12), 2605–2615. doi: 10.1109/TIFS.2015.2469254.

Smith, E. C. & Lewicki, M. S. (2006). Efficient auditory coding. *Nature*, 439(7079), 978–982.

Song, H. O., Girshick, R., Jegelka, S., Mairal, J., Harchaoui, Z. & Darrell, T. (2014). On learning to localize objects with minimal supervision. *Proceedings of the international conference on machine learning*.

Song, X., Jiao, L., Yang, S., Zhang, X. & Shang, F. (2013). Sparse Coding and Classifier Ensemble Based Multi-Instance Learning for Image Categorization. *Signal processing*, 93(1), 1–11.

Stikic, M., Larlus, D., Ebert, S. & Schiele, B. (2011). Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors. *Ieee transactions pattern analysis machine intelligence*, 33(12), 2521–2537.

Stone, M. (1974). Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the royal statistical society. series b (methodological)*, 36(2), 111–147.

Suckling, J., Parker, J., Dance, D., Astley, S., Hutt, I., Boggis, C., Ricketts, I., Stamatakis, E., Cerneaz, N., Kok, S. & Others. (1994). The mammographic image analysis society digital mammogram database. *Exerpta medica. international congress series*, 1069, 375–378.

Sun, Y.-Y., Ng, M. K. & Zhou, Z.-H. (2010). Multi-instance Dimensionality Reduction. *Proceedings of the aaai conference on artificial intelligence*, pp. 587–592.

Tang, K., Yao, B., Fei-Fei, L. & Koller, D. (2013). Combining the Right Features for Complex Event Recognition. *Proceedings of the international conference on computer vision*.

Tang, M., Luo, X. & Roukos, S. (2002). Active Learning for Statistical Natural Language Parsing. *Proceedings of the annual meeting on association for computational linguistics*.

Tapus, A. & Mataric, M. J. (2008). Socially Assistive Robots: The Link between Personality, Empathy, Physiological Signals, and Task Performance. *Aaai spring symposium on emotion, personality and social behavior*.

Tax, D. M. J., Hendriks, E., Valstar, M. F. & Pantic, M. (2010). The Detection of Concept Frames Using Clustering Multi-instance Learning. *Proceedings of the international conference on pattern recognition*.

Tax, D. M. & Duin, R. P. (2008). Learning Curves for the Analysis of Multiple Instance Classifiers. *Structural, syntactic, and statistical pattern recognition*.

Tax, D. & Cheplygina, V. (2015). MIL, A Matlab Toolbox for Multiple Instance Learning. version 1.1.0, Consulted at http://prlab.tudelft.nl/david-tax/mil.html.

Tjondronegoro, D. W. & Chen, Y.-P. (2010). Knowledge-Discounted Event Detection in Sports Video. *Ieee transactions on systems, man, and cybernetics - part a: Systems and humans*, 40(5), 1009–1024. doi: 10.1109/TSMCA.2010.2046729.

Tong, S. & Koller, D. (2001). Support Vector Machine Active Learning with Applications to Text Classification. *Journal of machine learning research*, 2, 45–66.

Tong, T., Wolz, R., Gao, Q., Guerrero, R., Hajnal, J. V., Rueckert, D., Initiative, A. D. N. et al. (2014). Multiple instance learning for classification of dementia in brain mri. *Medical image analysis*, 18(5), 808–818.

Tosic, I. & Frossard, P. (2011). Dictionary Learning. *Signal processing magazine, ieee*, 28(2), 27–38. doi: 10.1109/MSP.2010.939537.

Trigeorgis, G., Ringeval, F., Brueckner, R., Marchi, E., Nicolaou, M. A., Schuller, B. & Zafeiriou, S. (2016). Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. *Proceedings of the international conference on acoustics, speech, and signal processing*. doi: 10.1109/ICASSP.2016.7472669.

Uleman, J. S., Newman, L. S. & Moskowitz, G. B. (1996). People as flexible interpreters: Evidence and issues from spontaneous trait inference. *Advances in experimental social psychology*, 28, 211–280.

Vanwinckelen, G., do O, V., Fierens, D. & Blockeel, H. (2015). Instance-level accuracy versus bag-level accuracy in multi-instance learning. *Data mining and knowledge discovery*, 1–29.

Venkatesan, R., Chandakkar, P. & Li, B. (2015). Simpler Non-Parametric Methods Provide as Good or Better Results to Multiple-Instance Learning. *Proceedings of the international conference on computer vision*.

Ventura, S., Romero, C., Zafra, A., Delgado, J. A. & Hervas, C. (2008). Jclec: A java framework for evolutionary computation. *Soft computing*, 12(4), 381–392.

Veropoulos, K., Campbell, C. & Cristianini, N. (1999). Controlling the Sensitivity of Support Vector Machines. *Proceedings of the international joint conference on artificial intelligence*.

Vezhnevets, A. & Buhmann, J. M. (2010). Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Vijayanarasimhan, S. & Grauman, K. (2008, Jun). Keywords to Visual Categories: Multiple-Instance Learning For Weakly Supervised Object Categorization. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Vijayanarasimhan, S. & Grauman, K. (2014). Large-Scale Live Active Learning: Training Object Detectors with Crawled Data and Crowds. *International journal of computer vision*, 108(1), 97–114.

Viola, P., Platt, J. C. & Zhang, C. (2006). Multiple Instance Boosting for Object Detection. *Proceedings of neural information processing systems*.

von Melchner, L., Pallas, S. L. & Sur, M. (2000). Visual behaviour mediated by retinal projections directed to the auditory pathway. *Nature*, 404(6780), 871–876. doi: 10.1038/35009102.

Wagstaff, K. L. & Lane, T. (2007). Salience Assignment for Multiple-Instance Regression. *Proceedings of the international conference on machine learning*.

Wang, H., Ullah, M. M., Klaser, A., Laptev, I. & Schmid, C. (2009, Sep). Evaluation of Local Spatio-Temporal Features for Action Recognition. *Bmvc 2009 - british machine vision conference*, pp. 124.1–124.11. doi: 10.5244/C.23.124.

Wang, H.-Y., Yang, Q. & Zha, H. (2008a). Adaptive P-posterior Mixture-model Kernels for Multiple Instance Learning. *Proceedings of the international conference on machine learning*.

Wang, J., Li, B., Hu, W. & Wu, O. (2011). Horror video scene recognition via Multiple-Instance learning. *Proceedings of the international conference on acoustics, speech, and signal processing*.

Wang, J. & Zucker, J.-D. (2000). Solving the Multiple-Instance Problem: A Lazy Learning Approach. *Proceedings of the international conference on machine learning*.

Wang, Q., Si, L. & Zhang, D. (2012). A Discriminative Data-Dependent Mixture-Model Approach for Multiple Instance Learning in Image Classification. *Proceedings of the europeen conference on computer vision*.

Wang, X. & Zhang, X.-P. (2012). Ice Hockey Shooting Event Modeling with Mixture Hidden Markov Model. *Multimedia tools and applications*, 57(1), 131–144.

Wang, Z., Zhao, Z. & Zhang, C. (2016). Learning with only multiple instance positive bags. *Proceedings of the international joint conference on neural networks*.

Wang, Z. & Ye, J. (2015). Querying Discriminative and Representative Samples for Batch Mode Active Learning. *Acm transactions on knowledge discovery from data*, 9(3), 17:1—-17:23.

Wang, Z., Radosavljevic, V., Han, B., Obradovic, Z. & Vucetic, S. (2008b). Aerosol Optical Depth Prediction from Satellite Observations by Multiple Instance Regression. *Proceedings of the siam international conference on data mining*.

Warrell, J. & Torr, P. H. S. (2011). Multiple-Instance Learning with Structured Bag Models. *Energy minimization methods in computer vision and pattern recognition*.

Wei, X. S., Wu, J. & Zhou, Z. H. (2014). Scalable Multi-instance Learning. *Proceedings of the ieee international conference on data mining*.

Wei, X.-S. & Zhou, Z.-H. (2016). An empirical study on image bag generators for multi-instance learning. *Machine learning*, 1–44.

Weidmann, N., Frank, E. & Pfahringer, B. (2003). A Two-Level Learning Method for Generalized Multi-Instance Problems. *Proceedins of the european conference on machine learning*.

Wu, B., Zhong, E., Horner, A. & Yang, Q. (2014a). Music Emotion Recognition by Multi-label Multi-layer Multi-instance Multi-view Learning. *Proceedings of the acm international conference on multimedia*.

Wu, D. (2012). Genetic algorithm based feature selection for speaker trait classification. *Proceedings of interspeech*.

Wu, J., Zhu, X., Zhang, C. & Cai, Z. (2013). Multi-instance Multi-graph Dual Embedding Learning. *Proceedings of the ieee international conference on data mining*.

Wu, J., Zhu, X., Zhang, C. & Yu, P. S. (2014b). Bag Constrained Structure Pattern Mining for Multi-Graph Classification. *Ieee transactions on knowledge and data engineering*, 26(10), 2382–2396.

Wu, J., Pan, S., Zhu, X. & Cai, Z. (2015a). Boosting for Multi-Graph Classification. *Ieee transactions on cybernetics*, 45(3), 416–429.

Wu, J., Yu, Y., Huang, C. & Yu, K. (2015b). Deep multiple instance learning for image classification and auto-annotation. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Wu, J., Pan, S., Zhu, X., Zhang, C. & Wu, X. (2017). Positive and Unlabeled Multi-Graph Learning. *Ieee transactions on cybernetics*, 47(4), 818–829.

Wu, J., Hong, Z., Pan, S., Zhu, X., Cai, Z. & Zhang, C. (2014c). Exploring Features for Complicated Objects: Cross-View Feature Selection for Multi-Instance Learning. *Proceedings of the acm international conference on information and knowledge management*.

Wu, J., Zhu, X., Zhang, C. & Cai, Z. (2014d). Multi-Instance Learning from Positive and Unlabeled Bags. *Proceedings of the pacific asia knowledge discovery and data mining*.

Wu, R.-S. & Chung, W.-H. (2009). Ensemble one-class support vector machines for content-based image retrieval. *Expert system and application*, 36(3), 4451–4459.

Xiao, Y., Liu, B. & Hao, Z. (2016). A Sphere-Description-Based Approach For Multiple-Instance Learning. *Ieee transactions pattern analysis machine intelligence*.

Xu, D., Wu, J., Li, D., Tian, Y., Zhu, X. & Wu, X. (2017). SALE: Self-adaptive LSH encoding for multi-instance learning. *Pattern recognition*.

Xu, H., Venugopalan, S., Ramanishka, V., Rohrbach, M. & Saenko, K. (2016). A Multi-scale Multiple Instance Video Description Network. *Corr*, abs/1505.0.

Xu, X. & Frank, E. (2004). Logistic Regression and Boosting for Labeled Bags of Instances. In *Proceedings of the Pacific Asia Knowledge Discovery and Data Mining*.

Xu, Y. et al. (2014). Weakly supervised histopathology cancer image segmentation and classification. *Medical image analysis*, 18(3), 591–604.

Yan, S., Zhu, X., Liu, G. & Wu, J. (2016). Sparse multiple instance learning as document classification. *Multimedia tools and applications*, 1–18.

Yang, C., Dong, M. & Hua, J. (2006). Region-based Image Annotation using Asymmetrical Support Vector Machine-based Multiple-Instance Learning. *Proceedings of the ieee conference on computer vision and pattern recognition*.

Yu, G. & Slotine, J.-J. (2009). Audio classification from time-frequency texture. *Proceedings of the international conference on acoustics, speech, and signal processing*. doi: 10.1109/ICASSP.2009.4959924.

Yuan, L., Liu, J. & Tang, X. (2014). Combining example selection with instance selection to speed up multiple-instance learning. *Neurocomputing*, 129, 504–515.

Zafra, A., Ventura, S., Herrera-Viedma, E. & Romero, C. (2007). Multiple Instance Learning with Genetic Programming for Web Mining. *Computational and ambient intelligence*, 4507, 919–927.

Zafra, A. & Ventura, S. (2010). G3P-MI: A genetic programming algorithm for multiple instance learning. *Information sciences*, 180(23), 4496–4513.

Zafra, A., Pechenizkiy, M. & Ventura, S. (2012). ReliefF-MI: An Extension of ReliefF to Multiple Instance Learning. *Neurocomputing*, 75(1), 210–218.

Zafra, A., Pechenizkiy, M. & Ventura, S. (2013). HyDR-MI: A Hybrid Algorithm to Reduce Dimensionality in Multiple Instance Learning. *Information sciences*, 222, 282–301.

Zha, Z.-J., Hua, X.-S., Mei, T., Wang, J., Qi, G.-J. & Wang, Z. (2008, June). Joint multi-label multi-instance learning for image classification. *2008 ieee conference on computer vision and pattern recognition*. doi: 10.1109/CVPR.2008.4587384.

Zhang, B. & Zuo, W. (2008). Learning from Positive and Unlabeled Examples: A Survey. *Proceedings of the international symposium on information processing*.

Zhang, C., Chen, X., Chen, M., Chen, S.-C. & Shyu, M.-L. (2005). A Multiple Instance Learning Approach for Content Based Image Retrieval Using One-Class Support Vector Machine. *Proceedings of ieee international conference on multimedia and expo*.

Zhang, D., Wang, F., Si, L. & Li, T. (2011a). Maximum Margin Multiple Instance Clustering With Applications to Image and Text Clustering. *Ieee transactions on neural networks*, 22(5), 739–751.

Zhang, D., Wang, F., Shi, Z. & Zhang, C. (2010). Interactive localized content based image retrieval with multiple-instance active learning. *Pattern recognition*, 43(2), 478–484.

Zhang, D., Liu, Y., Si, L., Zhang, J. & Lawrence, R. D. (2011b). Multiple Instance Learning on Structured Data. *Proceedings of the 25th annual conference on neural information processing systems (nips)*.

Zhang, D., He, J. & Lawrence, R. (2013). Mi2ls: Multi-instance learning from multiple informationsources. *Proceedings of the acm international conference on knowledge discovery and data mining*.

Zhang, J., Marszalek, M., Lazebnik, S. & Schmid, C. (2006). Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *International journal of computer vision*, 73(2), 213–238. doi: 10.1007/s11263-006-9794-4.

Zhang, K. & Song, H. (2013). Real-time visual tracking via online weighted multiple instance learning. *Pattern recognition*, 46(1), 397–411.

Zhang, M.-L. & Zhou, Z.-H. (2004). Improve Multi-Instance Neural Networks through Feature Selection. *Neural processing letters*, 19(1), 1–10.

Zhang, M.-L. & Zhou, Z.-H. (2009). Multi-instance clustering with applications to multi-instance prediction. *Applied intelligence*, 31(1), 47–68.

Zhang, Q. & Goldman, S. A. (2001). EM-DD : An Improved Multiple-Instance Learning Technique. *Proceedings of neural information processing systems*.

Zhang, Q., Goldman, S. A., Yu, W. & Fritts, J. (2002). Content-Based Image Retrieval Using Multiple-Instance Learning. *Proceedings of the international conference on machine learning*.

Zhang, Y., Surendran, A. C., Platt, J. C. & Narasimhan, M. (2008). Learning from Multi-topic Web Documents for Contextual Advertisement. *Proceedings of the acm international conference on knowledge discovery and data mining*.

Zhou, Z.-h. (2004). *Multi-Instance Learning : A Survey*.

Zhou, Z.-H. & Xu, J.-M. (2007). On the Relation Between Multi-instance Learning and Semi-supervised Learning. *Proceedings of the international conference on machine learning*.

Zhou, Z.-H. & Zhang, M.-L. (2002). Neural networks for multi-instance learning. *Proceedings of the international conference on intelligent information technology*.

Zhou, Z.-H. & Zhang, M.-L. (2003). Ensembles of Multi-instance Learners. *Proceedins of the european conference on machine learning*.

Zhou, Z.-H. & Zhang, M.-L. (2007). Solving Multi-instance Problems with Classifier Ensemble Based on Constructive Clustering. *Knowledge and information systems*, 11(2), 155–170.

Zhou, Z.-H., Jiang, K. & Li, M. (2005a). Multi-Instance Learning Based Web Mining. *Applied intelligence*, 22(2), 135–147.

Zhou, Z.-H., Xue, X.-B. & Jiang, Y. (2005b). Locating Regions of Interest in CBIR with Multi-instance Learning Techniques. *Advances in artificial intelligence*.

Zhou, Z.-H., Sun, Y.-Y. & Li, Y.-F. (2009). Multi-Instance Learning by Treating Instances As non-I.I.D. Samples. *Proceedings of the international conference on machine learning*.

Zhu, J. Y., Wu, J., Xu, Y., Chang, E. & Tu, Z. (2015). Unsupervised Object Class Discovery via Saliency-Guided Multiple Class Learning. *Ieee transactions pattern analysis machine intelligence*, 37(4), 862–875.

Zhu, J., Wang, H., Yao, T. & Tsou, B. K. (2008). Active Learning with Sampling by Uncertainty and Density for Word Sense Disambiguation and Text Classification. *Proceedings of the international conference on computational linguistics*.

Zhu, J., Wang, B., Yang, X., Zhang, W. & Tu, Z. (2013). Action Recognition with Actons. *Proceedings of the international conference on computer vision*.