On the Improvement of Complexity Time and Detection Rate of Outlier Detectors: An Unsupervised Ensemble Perspective

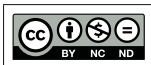
by

José Ramón PASILLAS DÍAZ

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

MONTREAL, "OCTOBER 24, 2017"

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC



BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS:

Mrs. Sylvie Ratté, Thesis Supervisor
Département de génie logiciel et des technologies de l'information, École de technologie supérieure

M. Mohamed Cheriet, President of the Board of Examiners
Département de génie de la production automatisée, École de technologie supérieure

Mr. Christian Desrosiers, Member of the jury
Département de génie logiciel et des technologies de l'information, École de technologie supérieure

Mrs. Louise Laforest, External Examiner
Département d'informatique, Université du Québec à Montréal

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON "OCTOBER 20, 2017"

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FOREWORD

This Ph.D. thesis was written during the period from winter 2015 until summer 2017. This a thesis based on articles required for the culmination of the Ph.D. program in engineering(profile applied research).

My interest in the field of outlier detection originated due to my participation in Kaggle competitions, which goal is to build the best possible model for the dataset at hand. The sources of the data varied widely depending on the application domain, but with some recurrent problems like unbalanced data, noisy attributes and presence of outliers; these outliers could represent only noise but also it could be a valid and even interesting observation. After searching in the literature for approaches for outlier detection, i observed that despite the importance of the field there was a lack of these kind of approaches, this scarcity was more evident in the ensemble setting. Therefore, i realized that an advancement on this field could provide researchers and practitioners in different domains with faster, more accurate and robust tools to reveal the outlier behavior hidden in the data.

Accordingly, the intent of this thesis is to design an ensemble approach for outlier detection that improves detection rate of a single algorithm while maintaining a lower execution time when compared with similar approaches present in the literature. Moreover, this approach should be able to detect outliers hidden deep inside the dimensionalty of the data.

This thesis encloses a succinct study of the outlier detection field, besides it addresses the impact that different distance measures have on an outlier detection algorithm or on an ensemble of these; based on which it proposes two unsupervised ensemble approaches for the detection of outlying observations.

ACKNOWLEDGEMENTS

I express my deepest gratitude to Prof. Sylvie Ratté, my thesis director, who with expertise and patience guided me into the intricate avenues of data mining and machine learning.

I thank to the board of examiners for accepting to review my thesis and providing me with valuable comments and suggestions. I wish to acknowledge Prof. Christian Desrosiers and Prof. Luc Duong who actively and patiently gave me invaluable feedback either in the form of comments after a presentation or by participating in the different evaluation stages of my studies.

I also thank all my colleagues from LINCS: Alpa Shah, Edgar García, Erick Velazquez, Laura Hernandez , Otilia Alejandro, Ruth Reategui and Kuldeep Kumar. Their constant feedback on my writing process left a deep influence in my professional writing style.

I also express my gratitude to my family: Fabiola Magallanes and Sophie Pasillas. Their support and company was critical and allowed me to focus on my Ph. D. defense. I would also like to thank my parents Ramón Pasillas and Herlinda Díaz by their, almost blind, trust.

I like to acknowledge the financial support of CONACYT México that made this project possible. Finally, i express my gratitude to the Universidad Autónoma de Zacatecas by their trust and support during these last years.

SUR L'AMÉLIORATION DE LA COMPLEXITÉ TEMPORELLE ET DU TAUX DE DÉTECTION DES DONNÉES ABERRANTES: UNE PERSPECTIVE D'UTILISATION DES MÉTHODES NON SUPERVISÉES FONDÉES SUR LES ENSEMBLES

José Ramón PASILLAS DÍAZ

RÉSUMÉ

Cette thèse présente deux algorithmes non supervisés pour détecter des données aberrantes dont le comportement est dissimulé dans des sous-espaces ou ne peut être identifié par l'utilisation d'un seul détecteur. Plus spécifiquement, nous examinons trois aspects : premièrement, la difficulté d'un seul détecteur à identifier différents types de valeurs aberrantes; deuxièmement, la propension des valeurs aberrantes intéressantes à se cacher dans des sous-espaces à faible dimension; troisièmement, l'impact des mesures de distance sur le processus de détection des valeurs aberrantes. Le but de cette thèse est d'améliorer notre compréhension des données dont le comportement aberrant n'est pas apparent, en utilisant des algorithmes simples de détection des valeurs aberrantes. En conséquence, nous avons abordé trois problèmes spécifiques. D'abord, nous proposons une méthode basée sur un ensemble de différents types de détecteurs dont les poids sont attribués de manière non supervisée. Ensuite, nous proposons un ensemble de détecteurs permettant d'identifier les observations dont le comportement aberrant est identifiable uniquement dans des sous-espaces spécifiques. Finalement, nous avons développé un schéma permettant de comprendre comment un seul détecteur ou un ensemble de détecteurs est influencé par la sélection d'une métrique de distance et son interaction avec différentes dimensions, tailles de données, paramètres ou composants d'ensemble.

Il existe de nombreux algorithmes permettant de détecter les valeurs aberrantes. Cependant, les approches fondées sur des ensembles non supervisés sont relativement limitées en nombre et sont principalement axées vers la détection d'un type spécifique de valeurs aberrantes. En conséquence, notre premier objectif est de détecter, de manière non supervisée, un type distinct d'observations périphériques. Nous proposons une approche capable d'utiliser la sortie de différents types de détecteurs, en attribuant des poids spécifiques à chaque détecteur en fonction d'une évaluation interne (non supervisée) de la capacité de chaque algorithme à traiter une série de données spécifiques. De plus, cette approche attribue un deuxième poids à chaque observation afin d'augmenter l'écart entre les valeurs aberrantes et les valeurs induites, améliorant ainsi le taux de détection des valeurs aberrantes. La principale contribution de ce travail est un ensemble de détecteurs, dont les composants peuvent être basés sur des hypothèses adaptées, avec un taux de détection des valeurs aberrantes amélioré par rapport aux approches similaires pour la détection des valeurs aberrantes. Comme c'est le cas pour plusieurs méthodes dans la littérature, notre approche présente un temps de traitement linéairement dépendant du nombre de composantes dans l'ensemble.

La deuxième partie de cette thèse se concentre sur la détection d'un type complexe de valeurs aberrantes, connu dans la littérature comme des valeurs aberrantes intéressantes; celles-ci ne

sont détectables que dans des sous-espaces spécifiques, contrairement aux valeurs aberrantes simples qui sont détectables dans l'espace complet. Notre première approche précédente étant incapable de détecter en un temps acceptable ce type de valeurs aberrantes, notre deuxième objectif concerne donc la détection de valeurs aberrantes de dimensions inférieures dans un temps efficace en termes de calcul. Nous proposons ici un ensemble non supervisé basé sur différents sous-espaces et sous-échantillons de données qui fournit non seulement un taux de détection plus élevé, mais qui s'avère aussi plus efficace que les approches d'ensemble similaires et, dans certains cas, supérieur au taux de détection des algorithmes spécifiquement adaptés aux données. Les principales contributions de ce travail sont la possibilité de détecter des valeurs aberrantes de dimensions inférieures et un temps de traitement amélioré.

La troisième partie de cette thèse étudie les interactions entre la métrique de distance choisie, les paramètres des algorithmes, la taille des données, la dimensionnalité et le nombre de composantes dans l'ensemble. Par conséquent, notre troisième objectif est d'améliorer notre compréhension des multiples facteurs influençant un algorithme de détection des valeurs aberrantes. Un ensemble d'expériences a été conçu pour évaluer à la fois le taux de détection et le temps de traitement. Les expériences couvrent un large éventail de scénarios de données synthétiques et réelles. Nos expériences de données synthétiques permettent des perturbations dans la taille et la dimensionnalité des données, alors que les données réelles permettent d'évaluer et de varier les paramètres d'un algorithme. À notre connaissance, il s'agit de la première évaluation, prenant en compte un ensemble complet de facteurs, principalement les mesures de distance, de l'influence de ces variantes sur l'efficacité d'un détecteur de valeurs aberrantes. Les résultats obtenus dans cette étude peuvent s'avérer une étape clé pour développer de nouvelles approches fondées sur des ensembles ou encore pour sélectionner les paramètres adéquats dans les approches existantes.

Mots clés: Valeurs aberrantes, ensemble, apprentissage non supervisé, données non balancées

ON THE IMPROVEMENT OF COMPLEXITY TIME AND DETECTION RATE OF OUTLIER DETECTORS: AN UNSUPERVISED ENSEMBLE PERSPECTIVE

José Ramón PASILLAS DÍAZ

ABSTRACT

This thesis presents two unsupervised algorithms to detect outlier observations whose aberrant behavior is hidden in lower dimensional subspaces or cannot be identified with the use of a single detector. In particular, we contemplated three facets: first, the difficulty of a single detector to identify different types of outliers; second, the propensity of interesting outliers to hide in low dimensional subspaces; third, the impact that distinct distance measures have on the outlier detection process. The ambition of the proposed algorithms is to improve our understanding about data observations whose outlier behavior is not evident using simple outlier detection algorithms. Accordingly, we addressed three specific problems. First, we propose to design an ensemble based on different types of outlier detectors with a set of weights assigned without supervision. Second, we propose an ensemble to identify observations whose outlier behavior is visible only on specific subspaces. Third, we develop a scheme to understand how a single detector or an ensemble of outlier detectors is influenced by the selection of a distance metric and its interaction with different dimensionalities, data sizes, parameter settings or ensemble components.

There is a wide availability of algorithms aimed at detecting outliers. However, the number of unsupervised ensemble approaches is limited and are mainly oriented towards the detection of a specific type of outlier. Accordingly, our first goal is to detect, in a unsupervised manner, distinct type of outlying observations. We propose an approach capable of using the output of different types of detectors, assigning specific weights to each detector depending on an internal evaluation (unsupervised) of the ability that each algorithm has on the specific dataset at hand; furthermore, this approach assigns a second weight to each data observation in order to increase the gap between outlier and inliers, further improving the outlier detection rate. The main contribution of this work is an ensemble of outlier detectors, whose components can be based on different assumptions, with an enhanced outlier detection rate when compared with similar single and ensemble approaches for outlier detection. Nonetheless, our approach exhibits a processing time linearly dependent on the number of ensemble components; this behavior is not exclusive of our approach, being instead prevalent in the ensemble outlier detection literature.

The second part of this thesis focuses on the detection of a complex type of outliers, known in the literature as interesting outliers, which are detectable only on specific subspaces of the data, on the contrary simple outliers are detectable on full dimensionality. Since our first approach was unable to efficiently detect this type of outlier, our second goal is the detection of lower dimensional outliers in a computationally efficient time. We propose an unsupervised ensemble based on different subspaces and subsamples of data which provides a higher detection rate and is computationally more efficient than similar ensemble approaches; in some cases, our

approach is even better to that of a single execution of a simple outlier detection algorithm. The main contributions of this work are the possibility of detecting lower dimensional outliers within an improved processing time.

The last section of this thesis is oriented towards the study of the interaction between distance metric, parameter settings, data size, dimensionality and number of ensemble components in determining the detection rate and processing time of an outlier detector. Hence, our third goal is to improve our comprehension about the multiple factors influencing an outlier detection algorithm. A set of experiments has been devised to evaluate both detection rate and processing time. The experiments cover a wide set of synthetic and real-world data scenarios. Our synthetic data experiments allow us to introduce perturbations in the size and dimensionality of the data, while real world data permits an evaluation of the effect of varying the parameter settings of an algorithm. To the best of our knowledge this is the first evaluation considering a complete set of factors, mainly distance metrics, influencing the effectiveness and efficiency of an outlier detector. The understanding achieved in this study can be a key step towards the development of new ensemble approaches or the selection and parameterization of existing ones.

Keywords: outliers, ensemble, unsupervised learning, unbalanced data

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ABREVIATIONS

AUC	Area Under the Curve
CONACYT	Mexican Council of Science and Technology
EDCV	Ensemble of Detectors by Correlating Votes
EDVV	Ensemble of Detectors by Variability Votes
ETS	École de Technologie Supérieure
FB	Feature bagging
FBSO	Feature bagged subspaces for outlier detection
k-NN	*k*-Nearest Neighbor Algorithm
LOF	Local Outlier Factor
MLE	Maximum likelihood estimation
NN	Nearest Neighbor
PCA	Principal Component Analysis
PR	Precision - Recall
ROC	Receiver Operating Characteristics
TN	True Negatives
TNR	True Negative Rate
TP	True Positives
TPR	True Positive Rate
UCI	University of California, Irvine

INTRODUCTION

0.1 Outlier detection

Our society is built around a set of predefined ideas about the expected behavior of the world; these ideas are related to the mechanisms with which our brain processes information. The human mind builds abstractions of all the objects and events that it encounters; however, a real object has to lose some of its characteristics during this abstraction process. Classification models mimic, to some extent, the human abstraction process, weighing heavily regular behavior. However, infrequent events, when present, can disturb and even deface our carefully constructed models. These events are usually known as outliers or anomalies, which have been defined as an observation or group of observations that deviate markedly from the remaining of the data. (Barnett & Lewis, 1994; Grubbs, 1969).

Differently from the classification field where the main aim is to build a model which characterizes the behavior of the majority of the observations, outlier detection focuses on those infrequent and outnumbered observations that could simply correspond to an error or noise in the data, but could also potentially portray a critical event of interest to the final user.

The impact of an outlying observation depends completely on the application domain. The application domains where outlier detection operates vary widely, e.g., breast cancer detection, fraud detection, intrusion detection, etc. It is important to note that different application domains usually require the detection of specific types of outliers which can be detectable using different types of algorithms, parameter settings or subsets of dimensions.

0.2 Problem statement

Outlier detection is a very challenging problem, which has not been fully solved. Despite the quantity and variety of approaches proposed in the literature, three problems remain unsolved.

First, a simple set of data can enclose multiple types of outliers and a single detector, being based on strong assumptions about what constitutes an outlier, is able to detect only deviations of a particular type. Second, it is very difficult for an outlier detector to find interesting outliers, which are predominantly hidden deep inside lower-dimensional projections of the data. This double detectors' blindness to distinct types of outliers hidden in lower dimensional projections remains an open question. Finally, derived from the previous two problems, the outlier detection literature also lacks the understanding of the impact that different distance measures have on outlier detection algorithms when interacting with different parameters settings, types of algorithm and data characteristics.

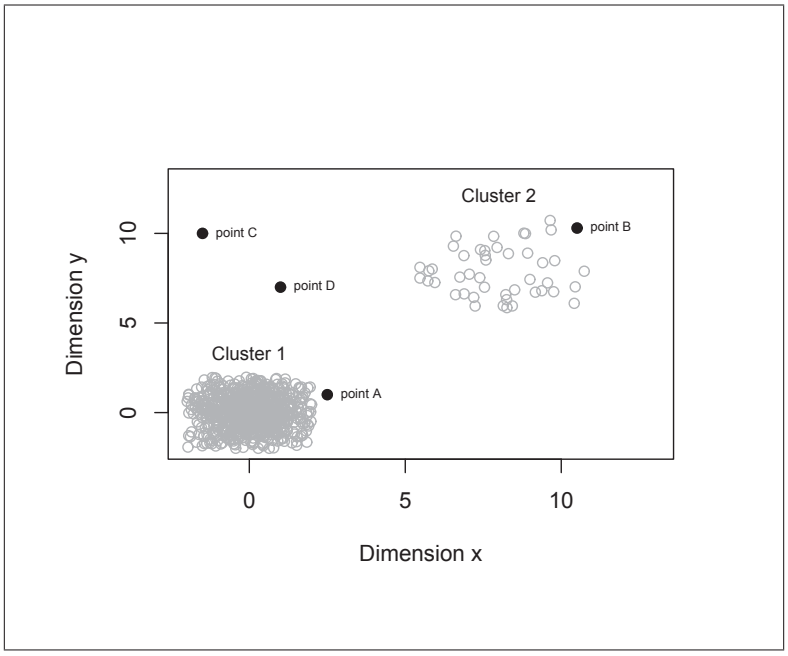


Figure 0.1 Outlier near clusters with different data density.

0.2.1 Diversity of outliers

There is a wide collection of approaches in the literature for outlier detection (Chandola *et al.*, 2009; Hodge & Austin, 2004). Each of these approaches is based on specific data assumptions

and is able to detect a precise type of outlier, namely proximity based (Breunig *et al.*, 2000), linear based (Piepel, 1989), statistical based (Laurikkala *et al.*, 2000), etc. In the context of the bias-variance trade-off, each outlier detection algorithm has an inherent bias towards a specific type of outlier, this is true even for detectors based on similar assumptions. For example, density-based and nearest neighbor algorithms, both of which are based on a related notion of similarity or proximity between observations, are usually unable to detect the same set of outliers, while density-based detector are capable to detect outliers located outside clusters with different densities, for nearest neighbor detectors this task results more challenging (Figure 0.1).

0.2.2 Hidden behavior of outliers

Despite the quantity and variety of approaches for outlier detection, most of them are capable of detecting outliers whose behavior is evident only on full dimensionality. However, observations whose outlier behavior can be revealed simply by using full data dimensionality are not an interesting case for outlier detection (Aggarwal, 2013a; Aggarwal & Yu, 2001; Zimek *et al.*, 2014), instead the interesting cases are those observations whose outlier characteristics are hidden on most subspaces (Figure 0.2 (a)), and are exposed only on specific but unknown subspaces (Figure 0.2 (b), 0.2 (c)); this type of outlier, albeit their high resistance to being detected, constitute the most interesting and challenging research path in outlier detection. In the classification field, ensembles of algorithms are usually used to improve the detection rate and robustness of a single classifier, yet in outlier detection this line of research has been scarcely investigated, with only a few approaches present in the literature.

0.2.3 Impact of distance measures

Most of the outlier detection literature is oriented towards the identification of outliers using some notion of proximity between observations. This type of approaches are usually evalu-

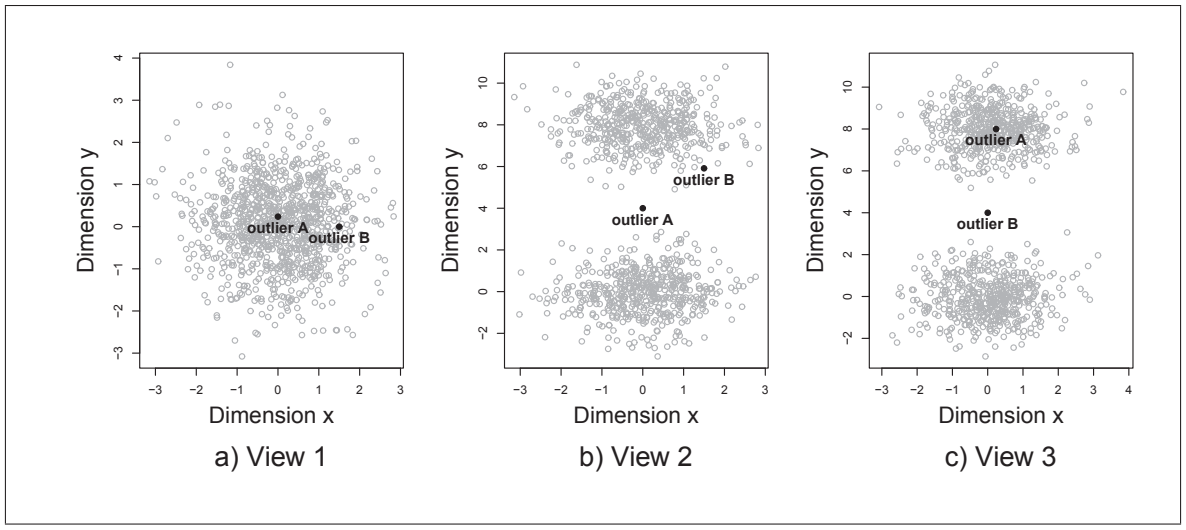


Figure 0.2 Outliers hidden in lower dimensional projections of the data. The figures represent the same set of data but plotted using different combinations of dimensions.

ated under a very limited set of configuration parameters, little is known about strengths and weaknesses of distinct distance measures when interacting with different types of data, dimensionalities, parameter settings, etc.

0.3 Objective and contributions

Past research in outlier detection well-established a large and diverse body of approaches oriented to the unsupervised outlier detection scenario. However, the fact that most of the existing approaches for outlier detection rely on specific assumptions of data, dimensionality or distance metric, is a challenge for the detection of diverse and interesting outliers. Accordingly, **the ambition of this thesis is to improve our understanding about data observations whose outlier behavior is not apparent using simple outlier detection algorithms**. Novel insights about these outliers can be critical mainly in unsupervised scenarios where there is a prevalent lack of information about the dataset at hand.

Three crucial considerations derive from the purpose of this thesis. First, the relative behavior of outliers depending on the application domain. Second, the difficulty of detecting observa-

tions whose outlier behavior is hidden in the lower-dimensional projections of the data. Finally, the interaction and impact that different distance metrics have on the outlier detection process.

The first part, **Chapter 2**, presents two mechanisms for combining the results of outlier detectors based on different assumptions. Both combination functions operate in an ensemble setting to localize outliers which could exhibit a disparate behavior. The use of different type of algorithms induces diversity in the ensemble, promoting a variance reduction, and hence increasing the detection rate of the algorithm. The proposed approach iteratively samples a user-specified number of subspaces, each of which contains a distinct set of dimensions with random lengths. The ensemble components are then, iteratively, applied over the random sets of dimensions producing a set of outlier scores for each algorithm of the ensemble; the combination functions are based on the dissimilarity and similarity between scores. Besides the mechanisms for scores combination, the approach also introduces the use of a set of capability of votes, distinct for each algorithm; the approach uses those votes as a way to weigh the potential ability of an algorithm over the particular dataset under study.

In **Chapter 3**, an unsupervised ensemble approach is proposed for the detection of outliers in high-dimensional data. This approach is able to detect outliers hidden in lower-dimensional projections of the data while operating in a lower execution time than similar approaches; this dual ability is the result of two distinct mechanisms used to induce diversity in the ensemble. Thus, this ensemble approach is able to detect interesting outliers which are only revealed in specific and unknown subsets of dimensions.

Finally, in **Chapter 4**, the behavior of different distance measures is analyzed using distinct data types, data dimensionality, data size and parameter settings. Furthermore, **Chapter 4** reveals the impact on the detection rate and processing time of different distance measures, proposing then, a guidance on the selection of distance measures for outlier detection.

0.4 Structure of thesis

The organization of this thesis is divided into five chapters (Figure 0.3). First, a review of the literature. Next, three proposed approaches. Then, a general conclusion.

Chapter 1 presents a background of the main concepts, methodologies and ensemble approaches in outlier detection. This section highlights the main limitations in current approaches for outlier detection.

Chapter 2 contains two novel mechanisms for a weighted combination of scores derived from different types of outlier detection algorithms. This work was published in a special issue in the journal Electronic Notes in Theoretical Computer Science (Elsevier).

Chapter 3 presents an ensemble approach for unsupervised outlier detection in lower-dimensional projections of the data. This chapter was published in the journal Computational Intelligence (Wiley).

Chapter 4 studies the impact on the detection rate and processing time of different distance measures when interacting with variations in parameters, algorithms and data. This study was submitted for publishing to the journal Information and Software Technology (Elsevier).

Chapter 5 summarizes all the work accomplished in this thesis, linking the outcomes in the different chapters, while highlighting their benefits and limitations.

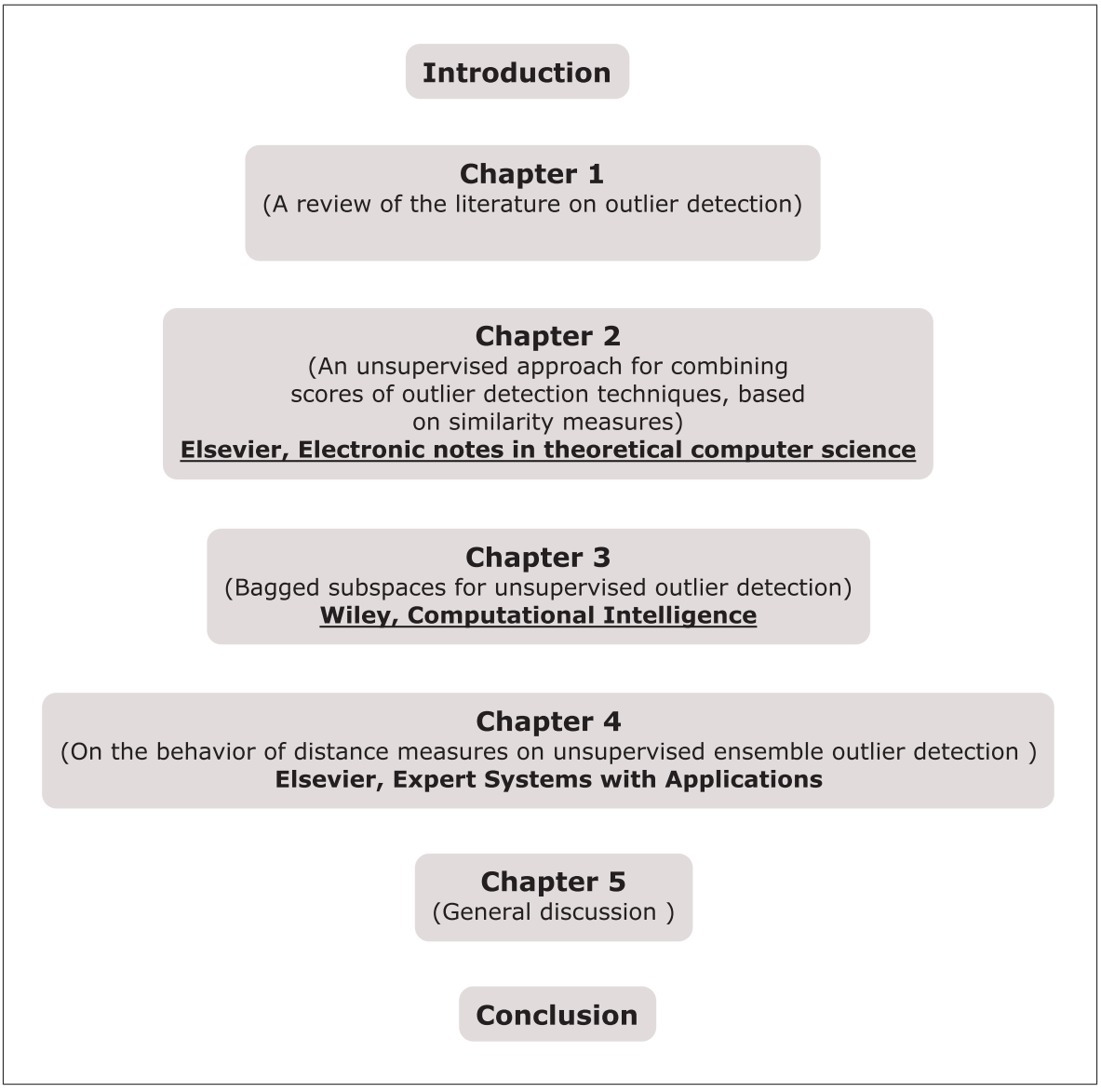


Figure 0.3 Structure of the thesis. Last line in bold and underlined indicates that the content has been published in a peer review journal. Last line in bold indicates that the content has been submitted to a peer review journal.

CHAPTER 1

LITERATURE REVIEW

One of the earliest modern definitions of outliers was made in 1969 by Grubbs who defined an outlier as: “An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs” (Grubbs, 1969). Later, in 1980 Hawkins defined an outlier as “An observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism” (Hawkins, 1980). Then Barnett and Lewis (Barnett & Lewis, 1994) improved these definitions by considering an outlier not only as a single observation but also as a group of observations inconsistent with the remainder of the data.

Despite the relative novelty of the field, the inherent characteristics of an outlier, like the sudden and critical impact that an undetected outlier could exhibit, have produced a diverse and vast outlier detection literature, with novel and interesting approaches proposed continually. Three comprehensive surveys summarizing the main approaches and their possible variations can be found in Chandola *et al.* (2009), Hodge & Austin (2004) and Zhang (2013). Furthermore, Aggarwal (2013b) introduced the first book fully dedicated to outlier detection.

The rest of this chapter is organized as follows: Section 1.1 reviews the diverse types of outlier detectors available in the literature, classifying them according to distinct criteria with which they were designed; Section 1.2 discusses advanced topics in outlier detection (E.g. high dimensionality, ensemble settings and a bias-variance trade-off) and how these problems have been approached in the literature; then, Section 1.3 reviews existing studies oriented towards the parameterization and evaluation of an outlier detection algorithm; finally, in Section 1.4, we highlight the limitations of the state of the art approaches for outlier detection.

1.1 Outliers heterogeneity

Despite the wide diversity of approaches for outlier detection (Table **??**), usually each field constrains the selection of an outlier detector depending on the specific context in which it operates, considering factors like the existence of class labels, specialized types of output or more importantly to those techniques commonly used in a specific application domain. However, outliers are not limited by such constrain, being possible to find distinct types of outliers coexisting in the same set of data; thus, the prevalent misconception of selecting a specific type of algorithm to detect the whole set of outliers in a datasets, could ignore those observations whose outlier behavior cannot be revealed by the selected technique.

1.1.1 Diversity of outlier detection algorithms

There are four main issues contributing to the current diversity of approaches in outlier detection:

- Ground truth availability. Outlier detection approaches used vary drastically depending on whether the data is labeled or unlabeled,
- Parameterization. The algorithm selection is also affected by whether or not there are some insights about the distribution of the data,
- Type of output. Outlier detection algorithms can also be selected depending on the type of output needed,
- Assumptions about the data. Since each outlier detection algorithm is based on strong assumptions about the data, their selection is application dependent.

Presence or absence of ground truth

Similarly to the classification field, outlier detection approaches can be classified according to the availability of ground truth. This means that the type of approach will depend of the presence or absence of labeled data; therefore, there are three approaches to outlier detection:

- Supervised
- Semi supervised
- Unsupervised

The absence of ground truth is not only a problem in the training phase, also the evaluation of an outlier algorithm is not possible due to the absence of labels. The same nature of the unsupervised scenario makes it difficult to have a straightforward evaluation as is the case in the supervised scenario. For this reason, the different algorithms available in the literature use datasets from the classification field, specifically imbalanced and labeled scenarios. This serves as a proxy that allows to use measures like ROC curves and AUC.

Parametric and not parametric approaches

Parametric approaches require knowledge about the data to analyze, assuming the data to follow a specific distribution (Barnett & Lewis, 1994; Rousseeuw & Hubert, 2011). Then, parametric approaches are best suited for scenarios where there is some prior knowledge about the statistical data distribution. In contrast, non-parametric approaches don't assume the data to follow a specific distribution and are more user independent, but they need some tuning, this can be in the form of k the number of neighbors for density-based techniques, number of centers for cluster-based techniques.

Type of output

Despite the multiple domains where it operates the final output an outlier detector is either in the form of binary decision or a degree of outlierness (Kriegel *et al.*, 2011). Then, there are two main types of outputs:

- Outlier scores (Knorr *et al.*, 2000; Jin *et al.*, 2001; Breunig *et al.*, 2000)
- Binary labels

Despite that the majority of outputs produced by existing outlier detection algorithms fall between the two previous categories, a third very compelling type of output has been increasing used in the literature, this output is based on probability estimates (Kriegel *et al.*, 2009a; Gao & Tan, 2006) that instead of providing a simple score, provides the estimated outlier probability of each observation. One of the main advantages of an outlier score over a simple outlier degree is that the former reveals much more information about the outlier behavior of the observation independently of a scale. An outlier degree can have widely different ranges depending on the outlier detection algorithm and the data at hand, this can also hinder their interpretability when used in an ensemble of outlier detectors.

Independently of the type of output, the outlying observations found in the detection process can be further categorized into relevant or not relevant (noise), the relevant concept usually depends on the application domain, as mentioned by Inatani & Suzuki (2002) "One person's noise is another person's signal". Unsupervised outlier detection naturally lacks the labels needed to train an algorithm using true examples of the true outlier class. However, despite this lack of information, most of the unsupervised detectors are able to return an output in the form of scores, such scores are usually in a spectrum ranging from normal data to outliers (Figure 1.1), while it is relatively less complicated to separate outliers from normal data, a straightforward separation between outliers and noise is a challenging task, usually with results where the possible outliers are contaminated with some noisy observations, or even some outliers missed among the noisy observations. Usually application domain knowledge is used to establish a threshold above which an observation is considered an outlier, also some approaches convert outlier scores into probability estimates which gain interpretability (Jing & Pang-Ning, 2006).

An outlier detection algorithm whose output is outlier scores instead of binary labels provides more insights about the outlier degree of an observation. However, in many domains it is also important the knowledge about why a particular deviations is behaving as an outlier, this concept known as "intentional knowledge" was first introduced by (Knorr & Ng, 1999). "An identified outlier should be explained clearly in a compact view, as a succinct subset of original features, that shows its exceptionality" (Dang *et al.*, 2014). The claim is that different outliers

Figure 1.1 Spectrum from normal data to outliers.
Image reproduced from (Aggarwal, 2013b).

can be hidden deep inside the dimensionality of the data, being observable in subspaces of the whole set of data. Then, the main idea of intentional knowledge is to find the smallest subspace where the outlier observations are located. The goal of intentional knowledge can add valuable information to the final user about an outlier, knowing not only what observations are outliers, but also an explanation about their outlier behavior. A few approaches have been proposed in the literature (Yang & Zhu, 2011; Huang & Yang, 2011; Angiulli *et al.*, 2009; Chen *et al.*, 2003; Marques *et al.*, 2015). Then having an outlier detection algorithm that provides outliers scores accompanied by their intentional knowledge greatly increases one of the main aspects of an algorithm, this is interpretability of results.

Data assumptions

Outlier detection algorithms are based on key assumptions about what constitutes an outlier; such restriction in the search space allows outlier detectors to specialize on a specific type of data, or more precisely, on a specific type of outlier. Specialized outlier detectors are then able to robustly detect a precise type of outliers, while overlooking non relevant or noisy observations, thus boosting detection rates while mitigating the number of false positives.

Highly data-specialized outlier detectors can exhibit blindness to unexpected types of outliers, such behavior can be present not only for observations beyond the outlier assumptions of the algorithms (Tan & Maxion, 2005), but also if the tuning parameters of the algorithm, for example the number of nearest neighbors in k-NN, are far from the optimal configuration for the specific data under study (Tan & Maxion, 2005). This selective blindness problem of outlier detection algorithms is far from trivial, selecting the wrong algorithm for a specific type of data results in a hopeless and faulty detection process; thus, the same algorithm that supposedly would unveil the outlier behavior in the data, is indeed biased against the type of outliers wanted, producing results near to a random guess or in the best case scenario detecting some outliers, but missing most of them.

The diversity of algorithms in outlier detection is, besides other factors, caused by the vast number of application domains (Aggarwal, 2013b). The following are some examples of such domains:

- Intrusion detection system.
- Credit card fraud.
- Loan application.
- Interesting sensor events.
- Manufacturing line fault detection.
- Satellite image detection.
- Medical diagnosis.
- Law enforcement.
- Earth science.
- Image novelty detection.

- Time series novelty detection.
- Text novelty detection.

1.1.2 Outlier detection algorithms based on specific assumptions

The diversity in domains where outlier detection operates results in a vast number of algorithms based on strong assumptions about the data. Next, we discuss how the different approaches for outlier detection are categorized according to the specific assumptions in which they are based. We also exemplify each category with iconic algorithms belonging to a specific type of algorithm.

Outlier detection algorithms can be classified, depending on their assumptions, broadly into 4 distinct categories; namely, extreme value analysis, probabilistic and statistical, linear models, and proximity based models. For ease of viewing the first 3 categories are grouped in Table 1.1 and the last category is depicted in Table 1.2.

Extreme values methods

Extreme value analysis represents the earliest and possible the simplest form of outlier detection. This type of method attempts to find those values that are found in the outskirts of a distribution. The basic, simple and indeed a rule of thumb is to declare as outlier those values 3 standard deviation above the mean (Knorr & Ng, 1997); such simplistic approach will obviously fail to detect an isolated point in the center of a set of points. The key step in this kind of method is to select an adequate distribution, thus, being able to detect the tails of distribution. Two major drawbacks of this approach are its reliance on a specific distribution and its limitation to work only on unidimensional data. The former refers to the characteristic of extreme value methods to depend on the right selection of a statistical distribution, in outlier detection the prevalent scenario is the lack of information about the data, thus complicating the selection of the optimal distribution. The latter addresses the characteristic of extreme value methods to work in a single dimensional space; there have been some approaches attempting to deal with

Table 1.1 Outlier detection methodologies (extreme value, probabilistic and linear models)

Type of algorithm	Advantages	Disadvantages
Extreme value analysis	- The higher the number of observations The most statistically representative it is. - Can be applied to the scores of most outlier detectors to transform the scores into binary labels by classifying extreme values as outliers. - Can be adapted under certain conditions to multi-dimensional data.	- Only work with quantitative real or ordinal data. - Only detect points on the outskirts of the data. - Designed for unidimensional data.
Probabilistic and statistical models	- Can be applied to almost any type of data. - No data normalization needed. - Easy to apply. - No need to store data as it is stores in a model.	- If there is a poor fit to the assumed distribution the model can miss true outliers or misclassify inliers. - Not advisable for high-dimensional datasets. - Assumes independence among attributes, if this assumption does not hold the model would fit the data poorly. - In general, lack of interpretability of results in terms of intentional knowledge. - As number of parameters increases so does over-fitting. - Not efficient in large datasets.
Linear models	- Good results if data is highly correlated. - Create a simple model over which outliers scores can be computed.	- Difficult to interpret in high dimensionality. - High complexity with relatively high dimensionality - Does not work with uncorrelated data. - Expect the data to be aligned in lower dimensional projections. - Does not work with clustered data. - Does not provide information about intensional knowledge. - Only work with numerical data.

Table 1.2 Outlier detection methodologies (proximity based methods)

Type of algorithm	Advantages	Disadvantages
Clustering based	- Does not assume any specific distribution. - Distances are computed simply to aggregates of observations. - Faster than other proximity based methods.	- Results would vary depending on the initial clusters selection. - A combination of different runs of the algorithm is needed in order to obtain more robust results. - Does not take into account the direction of the spread of a cluster. - The granularity of the algorithm is not optimal for outlier detection in small datasets.
Nearest neighbor based	- Does not assume any specific distribution - Can detect small groups of clusters	- Compute distances between each pair of observations, being computationally expensive. - Does not take into account regions with different densities.
Density based	- Does not assume any specific distribution - Can detect outliers taking into account regions with different densities. - Distance definition of outliers is easy interpretable. - Usually show higher detection rates than other proximity based methods.	- Compute distances between each pair of observations, being computationally expensive - Is largely affected by curse of dimensionality.

this limitation by considering multidimensional data. (Johnson *et al.*, 1998; Laurikkala *et al.*, 2000; Ruts & Rousseeuw, 1996), but often they lack the ability to detect the inter-attribute interactions when computing the deviation scores, furthermore, such approaches cannot undertake the inability of the basic approaches to detect outliers aside from those on the outskirts of the data.

Instead of using it as a regular outlier detector, extreme value analysis is usually used as the last step in an outlier detection process, as it can be applied to the scores produced by more sophisticated algorithms to transform outlier scores to binary labels.

Probabilistic and statistical methods

Similarly to extreme value detection, statistical methods assume a probability model which fit the underlying data. Indeed, extreme value methods can be considered a primitive and unidimensional form of probabilistic and statistical methods. This type of approaches declare as outliers those points that does not fit an assumed distribution.

Probabilistic methods are broadly classified, depending on their assumptions about the distribution of the data, into two categories: parametric and non-parametric.

Parametric

Probabilistic parametric methods assume a specific distribution of the data and learn the parameters of the model based on the training data (Barnett & Lewis, 1994; Eskin, 2000; Rousseeuw & Hubert, 2011). This type of approach can use Maximum likelihood estimation (MLE) to estimate the parameters of a Gaussian distribution, then conducting discordance tests to ascertain that the assumed distribution is close to optimal (Beckman & Cook, 1983; Barnett, 1976; Kamber *et al.*, 2012).

Non-parametric

This type of algorithm does not make any assumption about the underlying data distribution (Desforges *et al.*, 1998). Most of approaches can be further categorized as histogram or kernel

based. The former uses training data to construct a histogram for each feature, labeling as outliers the test observation that does not belong to any existing bin (Helman & Bhangoo, 1997; Javitz & Valdes, 1991), the histograms can also be built using only outlier data (Eskin, 2000; Dasgupta & Nino, 2000), then assigning any test instance falling into the existing bins as outlier. The latter type is based on kernel functions to find an approximate density distribution based on training data, a test instance is declared as outlier if it belongs to a low-density area of the distribution (Branch *et al.*, 2013; Palpanas *et al.*, 2003).

Linear methods

This kind of algorithms try to fit the data to an optimal hyperplane. Such hyperplane is usually determined by using least squares fit. In outlier detection, the outlier scores correspond to the distances of each point to the projected value in the hyperplane (Aggarwal, 2005; Arning *et al.*, 1996; Rousseeuw & Leroy, 2005), the larger the score the highest the assigned propensity of the observation to be an outlier. Then, such algorithms attempt to find a correlation or dependency of the dependent variable (Y) over the independent variables (X) in the form $Y|X$. Basically this type of method can work either in reverse or direct search (Zhang, 2013). The former, fits a linear model using all the data available, then assigning an outlier score equals to the square of the residuals between each point and its projected value in the hyperplane. The latter, fit a linear model using only a portion of the data and then, incrementally, it adds more values which exhibit the lowest deviation from the hyperplane, the remaining observations in the last iteration of the algorithm are those exhibiting the largest deviation from the projected hyperplane, having in consequence the largest outlier score. PCA (Jolliffe, 2002) is a related method that can be used for outlier detection by projecting the data into a lower dimensional subspace, then predicting values of all observations by projecting them into the principal components, being outliers those points whose actual and predicted value differ (Korn *et al.*, 1997).

Proximity based methods

Proximity based methods use distances and/or density estimation to define the outlier score of an observation, being outliers those points which are isolated from the remaining observations.

Proximity based methods are strongly based on the computations of similarities or distances between observations, thus defining an appropriate distance metric is a critical step in this class of algorithms (see Section 1.3). Proximity based methods are the most popular type algorithms in the outlier detection literature, mainly due to their simplicity and absence of assumptions related to the underlying data distribution. Similarity is a relative concept that depends on the interpretation of proximity used. Such proximity can be computed using three main methods: nearest-neighbors, densities or clusters.

Clustering based

Clustering based methods for outlier detection (Eskin *et al.*, 2002; Khoshgoftaar *et al.*, 2005; Muller *et al.*, 2012b; Ng & Han, 1994; Zhang *et al.*, 1996) are based on the idea, borrowed from the classification field, of cluster detection, the aim is the detection of dense groups of points by assigning each point to a specific cluster; measuring the fit to a cluster is usually done by computing the distances of each point relative to the centroid of all available clusters, an observation is then assigned only to the cluster whose centroid is close. Outliers are reported, in most cases, as a side product of the process, as those points which do not belong to a cluster and using their relative distances to the nearest centroid as outlier scores.

The results of this kind of approach can vary between different runs of the algorithm depending on the initial setup of clusters, also the quantity of clusters (k) is a user specified parameter; being outliers reported as a side product of the clustering process, clustering algorithms often fail to detect outliers which are grouped in small clusters. With prior knowledge about the outlying observations in the data it is possible specify a convenient k to detect even outliers grouped in clusters, however, being outlier detection essentially a prevalent unsupervised problem, the heuristic specification of k tend to produce not optimal results. Multiple iterations of the algorithm and the posterior combination of their outputs are often needed in order to obtain more robust results.

Nearest neighbor methods

Nearest neighbor methods are based on the measurement of distances between observations (Knox & Ng, 1998; Ramaswamy *et al.*, 2000) by using metrics like Euclidean distances (see Section 1.3). A use specified parameter k is used in order to determine the number of nearest neighbors to examine, being outliers those points with the higher scores computed by averaging the distance of the point to its k nearest neighbor.

In clustering methods once the centroid of each cluster is established, it is possible to measure the distance of a new instance only relative to the centroid of the data. Nearest neighbor methods compute distances between all instances in the data, having a higher level of granularity to that of clustering based methods. However, this richer granularity is accompanied for high-processing time, as the pairwise distance between any observations in the data needs to be computed, thus exhibiting a scaling quadratic processing time ($O(n^2)$).Different methods can be used in order to prune some points or portions of the space to reduce the amount of distance computations needed (Angiulli & Pizzuti, 2002; Eskin *et al.*, 2002; Wang *et al.*, 2005).

Density-based

Density methods are based on the same principles that clustering and nearest neighbor methods; however, besides the computation of distances between points, this type of algorithm weights such distances by using the densities of its k nearest neighbors, in this way an observation receives a high outlier score depending on its distance to its k neighbors and the relative density in which the observation and its k neighbors are located (Breunig *et al.*, 1999, 2000; Papadimitriou *et al.*, 2003). Density-based approaches are probably the most popular type of algorithms used in the literature, mainly due to their capability to identify outliers using local densities, their unsupervised nature, their instability depending on variations in the search space[1] and finally the relative simplicity of the local density.

Density-based algorithms are able to detect outliers in data with different densities, where clustering and distance methods will struggle. For example, in Figure 1.2 we plotted a synthetically

[1] Instability in the base algorithm is an interesting asset in an ensemble setting, as ensemble components with the same point of view do not provide gains to the ensemble, but complementary views of the data can offer significant gains when combined

created small dataset, the data consists of two main clusters, clusters 1 and 2, and points A, B, C, and D scattered around the main clusters. Points B and C are clearly outlying points lying far from the two clusters of the data, thus, any algorithm based on proximity can easily label them as outliers. However, points A and B portray a more challenging scenario, both points have a similar distance to their nearest neighbors, but the density of the nearest neighbors of point A is higher than that of point B, under this scheme point A should be considered an outlier, while point B as a simple inlier. An algorithm not considering local densities will strive to correctly labeled both of them correctly. In this kind of scenario lies the capability of density-based methods. Using local densities, a simple algorithm like Local outlier factor (LOF) can easily correctly label both points A and B as outliers and non-outliers, respectively.

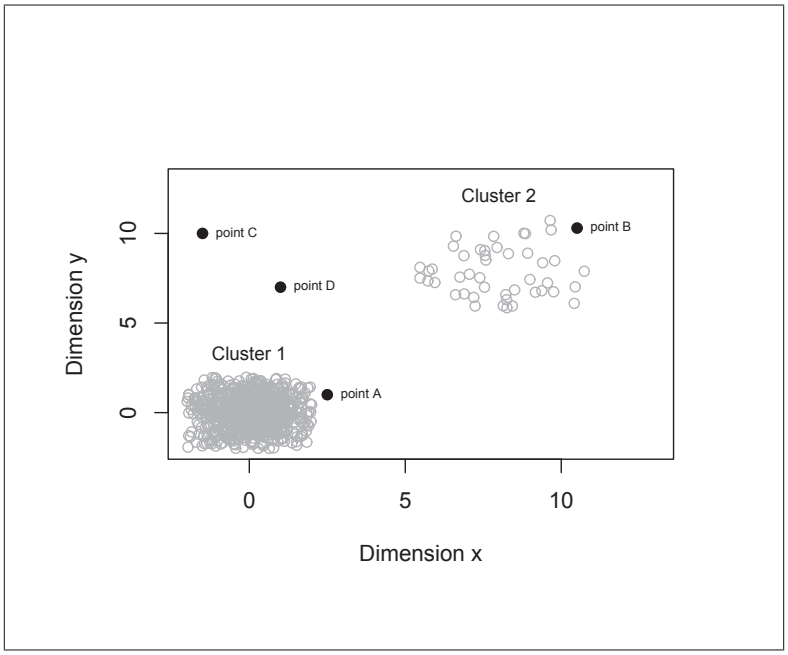


Figure 1.2 Local densities in density-based methods.

One of the most popular outlier detection algorithms in the literature is LOF (Breunig *et al.*, 2000), with different variations proposed in the literature (Jin *et al.*, 2006; Tang *et al.*, 2002), also the work in (Jin *et al.*, 2001) limit the search of LOF to only the top *(*n) outliers in the data. This approach is claimed to be able to detect local outliers located on different data densities (Eq. 1.1). LOF is able to capture local densities in the data by using a local reachability

between points (Figure 1.3). However, it has been argued in (Aggarwal & Sathe, 2015) that a simple average k-NN method can outperform LOF, in part due to the typical binary scenario of outlier detection (outlier or not outlier), being then the interesting outliers global in nature, also the LOF algorithm can have a bias due to its harmonic normalization capturing the noise in the data.

LOF requires a single parameter *MinPts* or k, which is the number of closest neighbors used to determine the neighborhood of an observation /texitp. The neighborhood of p are those observations with a distance least or equal that the distance to the k nearest neighbor. The number of points in different neighborhoods can be different due to ties in distances. LOF not only uses the reachability of p to k, but it also uses the reachability of each point in the neighborhood of p to its own k nearest neighbors. Thus, the lower the density of p and the larger the density of its k neighbors the higher the outlier scores assigned by LOF. In this way, LOF is able to assign larger scores to points depending on their relative isolation with respect to local neighborhoods in the data. E.g. the point p in the center of Figure 1.3 has a relative much lower density when compared with that of its neighbors.

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(p)}{lrd_{MinPts}(o)}}{N_{MinPts}(p)} \tag{1.1}$$

$$reach - dist_k(p,o) = max\{k - distance(o), d(p,o)\} \tag{1.2}$$

$$lrd_{MinPts}(p) = 1/(\frac{\sum_{o \in N_{MinPts}(p)} reach - dist_{MinPts}(p,o)}{N_{MinPts}(p)}) \tag{1.3}$$

It has been claimed in (Aggarwal & Sathe, 2015) that a simple averaged k-NN algorithms can outperform the iconic LOF(Aggarwal & Sathe, 2015), the authors in (Aggarwal & Sathe, 2015) show through different data scenarios, sample rates and parameter settings the differences in performances between LOF and averaged k-NN, and the overall gains when both algorithms are

used as base detectors in an ensemble setting. The authors in (Aggarwal & Sathe, 2015) argued that the main factor for the superior performance of averaged k-NN over LOF is due to the harmonic normalization used in LOF (Eq. 1.2, Eq. 1.3) , which captures the noise from dense regions in the data, this makes LOF an unstable algorithm which is indeed an interesting quality in an ensemble setting for variance reduction. However the intrinsic bias of LOF in its harmonic normalization degrades the ensemble performance more than the gains that can be achieved by the variance reduction. Accordingly, in our experiments with different density measures we will use both algorithms as base detectors (as done by Aggarwal (Aggarwal & Sathe, 2015)).

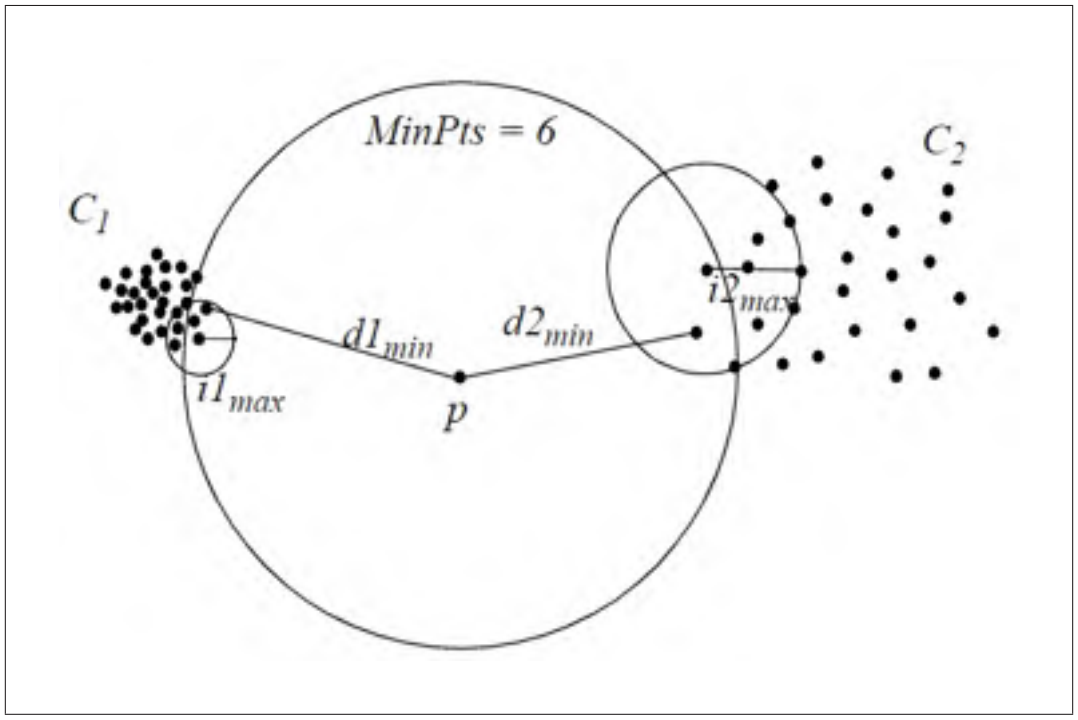


Figure 1.3 Local densities of point p and local densities of its nearest neighbors. Image reproduced from (Breunig *et al.*, 2000).

Clustering based

Clustering based methods for outlier detection are based on the strong assumption that outliers can be identified by their distance to a cluster or the size of the nearest cluster (Agrawal *et al.*, 1998; Ester *et al.*, 1996). However, this simplistic view results in the detection of noise or

weak outliers. Differently from other proximity based methods a clustering algorithm is able to compute outlier scores without using a pair-distance between al observations of the data, instead it computes only distance to the closest centroid, this leads to significant reduction in processing time at the expense of losing detail in a local analysis. Small clusters of outliers can be wrongly classified as inliers if the approximate number of clusters is unknown.

1.1.3 Combination functions

Different outlier detection algorithms produce scores which are not directly comparable, either by the type of output produced or by the scales of the scores. For example, ABOD (Kriegel *et al.*, 2008)(an outlier detection algorithm based on angles and distances between points) produces scores where the higher values correspond to lower outlier degree, but other algorithms like LOF (Breunig *et al.*, 2000) denote the outlierness degree with higher values. These two previous examples produced values with different ranges and with no upper value. However, one of the seminal algorithms for outlier detection DB-outlier (Knox & Ng, 1998) produces scores limited to the range [0,1].

There is no consensus on which combination function is best, while (Keller *et al.*, 2012) Zimek *et al.* (2014) argues that an average of the outlier results is better; while Aggarwal (2013a) argues that a maximum function avoids the dilution of scores and instead highlights observations with high outlier scores even on only some of the ensemble members. Aggarwal (2013a) argues that the average function will dilute the outlier behavior of some points whose behavior could have been highlighted only on some set of scores .Zimek *et al.* (2014) Disagree that the problem with the maximum function is due mainly that a single score that is far beyond the rest of the conglomerate of scores will decide the final decision , irrespective of the majority of the scores with similar opinion. In our personal opinion, both approaches to score combination have pros and cons, while it is true that in a dataset where all dimensions contribute to the classification of the outlier observations (absence of noisy attributes) the average function provides indeed an advantage by taking a consensus of in theory accurate and diverse results, if we consider outlier detection scenarios where the outlier behavior is buried deep inside the dataset

and is only visible using a specific set of dimensions, then the maximum function would be better, as it is capable of detecting that single result based on a subset of dimensions where the outlier observation is visible. We conclude in spite of the promising behavior of the maximum function, in reality it is very difficult to detect the complete set of attributes where a particular outlier is present, being the most common scenario to have different set of outlier scores obtained from subsets of attributes that can or cannot contain some of the relevant attributes, then the average of scores will make more sense by compensating the different error using a consensus of imperfect but hopefully slightly accurate classifiers.

This problem in interpretability of scores was brought to the attention by the authors of in citep-Kriegel2011. While some of the earliest approaches like DB-outlier produced comprehensible scores in the range [0,1], variations from this basic approach like LOF (Breunig *et al.*, 2000) or ABOD (Kriegel *et al.*, 2008) produce scores whose scale has no limits, then making impossible, without expert knowledge, to determine the true outliers. Then an important task to have in consideration is to transform the outlier scores of these types of algorithms to a probability estimate, which will provide a better interpretability for the final user. Thus, the ideal outlier score is regular and normal (Kriegel *et al.*, 2011), regular if $S(o) \geq 0$, $s(o) \approx 0$ for inliers and $s(o) \gg 0$ for outliers.

There is a variety of papers that discussed the problem of score comparability: Using sigmoid functions and producing as final output probability estimates (Gao & Tan, 2006) , simply transforming into standard deviations (Nguyen *et al.*, 2010), and the third one attempts to approximate the distribution of scores produced by different types of outlier detection algorithms, tailoring the scores depending on a specific distribution (Kriegel *et al.*, 2011).

The basic principle in transformation is that the ranking of the scores should not be inverted after the transformation. In Kriegel *et al.* (2011) several approaches for data normalization and regularization are proposed, where regularization basically transforms an outlier score into the range [0,Inf) and then normalization brings the scores to the scale [0,1]. It is important to

consider that the approaches proposed in Kriegel *et al.* (2011) can be applied as a post step to the scores produced by different outlier detection algorithms.

1.2 Hidden outlier behavior

In the previous chapter, "Outlier heterogeneity", we discussed the different types of outlier detection algorithms and how they are strongly tied to strong assumptions about the data. However, despite the complexity of the problems depicted in the previous chapter, a far critical, complex and interesting problem remains. Having considered that in a single dataset can coexist different types of outliers and that it is infeasible to capture all of them by using a single technique, we further need to contemplate that, at least in the interesting scenarios (Keller *et al.*, 2012), an outlier is usually located only in a specific subset of dimensions of a high-dimensional and unbalanced dataset. This limits the applicability of most of the current approaches in the literature, a blind use of an algorithm not adapted to this scenario will results in an inability to detect these interesting outliers.

1.2.1 The challenges of high-dimensional data

High-dimensional data is a challenging problem not limited to outlier detection, fields like classification (Kriegel *et al.*, 2009b; Domeniconi *et al.*, 2004; Parsons *et al.*, 2004) also struggle to find optimal solutions to this problem. The high-dimensional scenario is an evolving problem present not only in outlier detection, but also on classifications and clustering. Early papers on outlier detection considered $D \geq 5$ as a large dimensionality dataset (Knorr *et al.*, 2000), while current ensemble approaches need to deal with thousands and even larger numbers of dimensions originated from the increasing capacity of the systems to produce, recompile and store data. Then, the high-dimensional problem is far from being considered as solved, instead new approaches need to incorporate mechanisms to deal with this increasing dimensionality.

Accordingly, there are two main issues present in high-dimensional outlier detection:

- In high dimensionality all the points become almost equidistant to each other.

- Interesting outliers are usually located in a lower dimensionality of the data.

Sparsity of points

Outliers are usually located in sparse regions of the data, Aggarwal (2013b) describes a sparse region as "an abnormal lower dimensional projection is one in which the density of the data is exceptionally lower than average". One problem with points in high-dimensional data is that they are almost equally equidistant (Hinneburg *et al.*, 2000; Beyer *et al.*, 1999; Aggarwal & Yu, 2001); thus, as the number of dimensions increases so does the distance between the points. If each point in the data space is located in a sparse region then all points are erroneously considered as outliers. However, Zimek *et al.* (2012) points out that the concentration effect is not the main problem in high-dimensional outlier detection; Zimek argues that as the number of relevant attributes increases then the concentration effects are diluted, and instead , the outlier behavior is more obvious , and it keeps doing it increasing even more the dimensionality, Zimek states that “For points that deviate in every attribute from the usual data distribution, the outlier characteristics just become even stronger and more pronounced with increasing dimensionality”. There is a bias of some type of outlier detection algorithms towards high dimensionality datasets, tending to assign higher scores as the dimensionality of the data increases.

Outlier located in low dimensional projections

The dimensionality in which an outlier is located determines the level of complexity required in the outlier detector. A detector limiting its search space to a single dimensional analysis, ignoring the relationships between attributes, is able to detect only trivial outliers (Keller *et al.*, 2012), in contrast non-trivial outliers or interesting outliers, the most challenging and critical type of outliers, are usually located in specific subspaces of the data, their outlier behavior is not commonly exhibited in a single dimension, but instead it is revealed only in a specific combination of dimensions. An example is a 20-year-old patient with cancer (a typical outlier as the combination of age and cancer is not common). The age 20 or the presence of cancer are not too uncommon to be considered an interesting outlier. Thus, analyzing these attributes individually does not provide any insights about a potential outlying behavior. A simple extreme

value detector, which assumes outliers located in tails of a distribution, focused on individual and independent attributes, will fail to detect the 20-year-old cancer patient. Nonetheless, an analysis of the previous example but using both features unveils that such a combination of age and presence of cancer is sufficiently anomalous as to be considered an outlier. There are multiple supervised solutions for this lower dimensional problem; however the lack of ground truth labels in its unsupervised analog depicts an interesting challenge. Moreover, the previous example depicted a scenario where the outlier behavior was observed in full dimensionality; however, in most real-world scenarios interesting outliers are located in high-dimensional datasets and their outlier behavior is only observable in specific subspaces of the data. Thus, interesting outliers are neither located in individual dimensions nor in full dimensionality.

High-dimensional data impedes a blunt search for outliers based in all the possible combinations of attributes, the processing time increases exponentially as the dimensionality of the data rises. Besides dimensionality, the size of the dataset, also plays an important role to determine the processing time of a detector; however, Filzmoser *et al.* (2008) argues that "Computation time increases more rapidly with p than with n", here p stands for dimensionality, whereas n is the number of observations. Using a brute force search process throughout all possible subsets of attributes is infeasible, the quantity of spaces to analyze is $2d$-1, in low-dimensional data this does not represent a problem, but as the dimensionality of the data increases the challenge becomes more evident. E.g. if d=2, the number of subspaces are 2^2=4, but even in a relatively modest dimensionality of 10, the number of subspaces to analyze rises to 2^{10}=1024.

An optimal solution for the detection of outliers in lower dimensional subspaces is to specifically select the relevant dimensions to be used in the analysis, however, outlier detection is constricted by the inherent unsupervised nature of the process, complicating an otherwise straightforward picking of the most contributing and most relevant attributes. Moreover, in outlier detection the number of useful dimensions is often very limited, then wrongly omitting a few of the contributing features could inadvertently cause more damage than that caused by including some irrelevant dimensions in the process. A more viable approach could be to use random sets of attributes (Hawkins, 1980; Keller *et al.*, 2012) . Differently from classification,

feature selection in outlier detection is very difficult as it is not possible to use robust statistics to select the relevant dimensions where a specific outlier is located, “Robust statistics is all about more data, and outliers are all about less data and statistical nonconformity with most of the data!” (Aggarwal, 2013b).

Despite these challenges, some approaches like HICS (Keller *et al.*, 2012) attempt to find those subspaces with high contrast and a strong correlation, ignoring those subspaces with low contrast, which potentially can result in a low-dimensional data with relevant attributes. However, if relevant attributes are missing then the selection process is irremediably biased.

Campos *et al.* (2015) proposed an evolutionary approach to tackle the high-dimensional scenario. The final set of outlier scores are computed in a set of dimensions selected by a process imitating natural selection, where only the fittest of the solutions (sets of projections with a density which is lower than average) survive to next phase of the algorithm, random mutation of some parameters of the solutions is used to induce variability and diversity of solutions in the selection mechanism. As the process progress the set of solutions become more and more similar converging to an, in theory, optimal solution. Despite the appealing approach of evolutionary search algorithms imitating natural processes, this algorithm has a main drawback, the outlier detection problem by definition is in general characterized by the lack of information about the dataset at hand and evolutionary algorithms need advanced domain knowledge of the data under study, this characteristic of evolutionary algorithms makes their use in unsupervised outlier detection not infeasible, but at least circumscribed to the existence of some domain insights. A method that uses principal components is proposed in (Filzmoser *et al.*, 2008), here outliers are identified in the projected space that conserves only those components that represent a level of the total variance.

Instead of attempting to select the right set of relevant dimensions for each set of outliers, Lazarevic & Kumar (2005) proposed an approached named feature bagging which selects randomly n different sets of attributes, this mechanism improves detection rate by combining diverse sets of results. The authors in (Lazarevic & Kumar, 2005) propose two mechanisms

to combine the outputs obtained by each component in the ensemble, namely *Breadth-First* and *Cumulative Sum*. The former sorts the scores from each detector in descending order, then selects the indexes of the largest ones as an outlier degree, this is equivalent to a combination function where the maximum score is selected from the results of each detector. The latter is a simple average of the results. Both approaches are reported to achieve detection rates superior to that of the base detector; however, the averaging procedure results more appealing due to its inherent capacity to reduce the global variance of the ensemble due to the diverse set of results in which it is based.

1.2.2 Accuracy and diversity

An important point when constructing an ensemble is to have members that individually perform better than random guessing and whose errors are uncorrelated (Opitz & Maclin, 1999; Chandra *et al.*, 2006), these correspond to accuracy and diversity, respectively . As mention in Tan & Maxion (2005) even in the case of using different types of detectors, these can be blind to the same regions in the data space, the main reasons for this blindness could be the inability of the outlier detection algorithm to detect a specific type of anomaly, an incorrect parameterization of the algorithm or wrongly setting, too low or too high, the threshold to flag an observation as an outlier.

In a supervised scenario measuring accuracy is a relatively straightforward task, as it is possible to use the ground truth classification of each observation to compute measures like accuracy. However, the lack of labeled data and the extremely low proportion of outliers limits the types of evaluation methods that can be used in outlier detection; nonetheless, Section 1.3 presents some evaluation methods that can be used in outlier detection.

Diversity and accuracy are two concepts that in an ensemble settings are dependent, as highly diverse classifiers tend to produce improvements in the detection rate in an ensemble setting. Each algorithm searches the best possible hypothesis among the space of possibilities H (Chandra *et al.*, 2006; Ditterrich, 1997). Combining different hypotheses can provide a good approx-

imation of the true but unknown hypothesis. Even finding the best hypothesis has been considered as NP-complete problem (Blum & Rivest, 1989). Uncorrelated results, when combined, tend to produce positive detection improvements, and correlated results produce lower or in some cases negative gains (Schubert *et al.*, 2012).

In the unsupervised scenario, the true output or the ideal hypothesis is usually not comprised in the space of results in the ensemble, and instead an approximation could be obtained from the set of available models, and in this way have the best possible hypothesis for the current model and available data. E.g. in Figure 1.4 the solid circle represents the ground truth, the red crosses are the results from individual classifiers, and the average of the outputs from each classifier is represented as a solid triangle, in the first scenario (Figure 1.4 (a)) the scores provided by each detector do not contain the true classification, but the diversity in their results allows to produce a result that is approximately closer to true value. In the second case (Figure 1.4 (b)) the diversity in the detectors is diminished and are further hinder by their biased behavior towards relatively high values, the result is that even after combination, the final output is wrongly assigned due to the lack of diversity and biased results in the individual detectors. In the two-dimensional scenario in Figure 1.5 each axis (x,y) represents the scale in the scores provided to two distinct objects, this pair of objects is iteratively scored by different pairs of detectors, the circle represents the ground truth, red crosses display the results from different classifiers, in Figure 1.5 (a) the diversity in the classifiers results in a combined output which is closer to the ground truth than that of any of the individual classifiers; however, in Figure 1.5 (b) the individual scores are partially and wrongly concentrated distantly from the ground truth object, thus highly concentrated and inaccurate detectors would invariably hinder the detection process; moreover, not knowing the ground truth output it is impossible to use measures like accuracy, this suggests that without control over the individual accuracy in the detectors, diversity should be induced to cover a wider search space and as the individual results are combined obtain an improved detection rate in expectation.

There are mainly 5 methods for inducing diversity (Zimek *et al.*, 2014): by varying the set of dimensions or attributes (Lazarevic & Kumar, 2005; Keller *et al.*, 2012), by subsampling

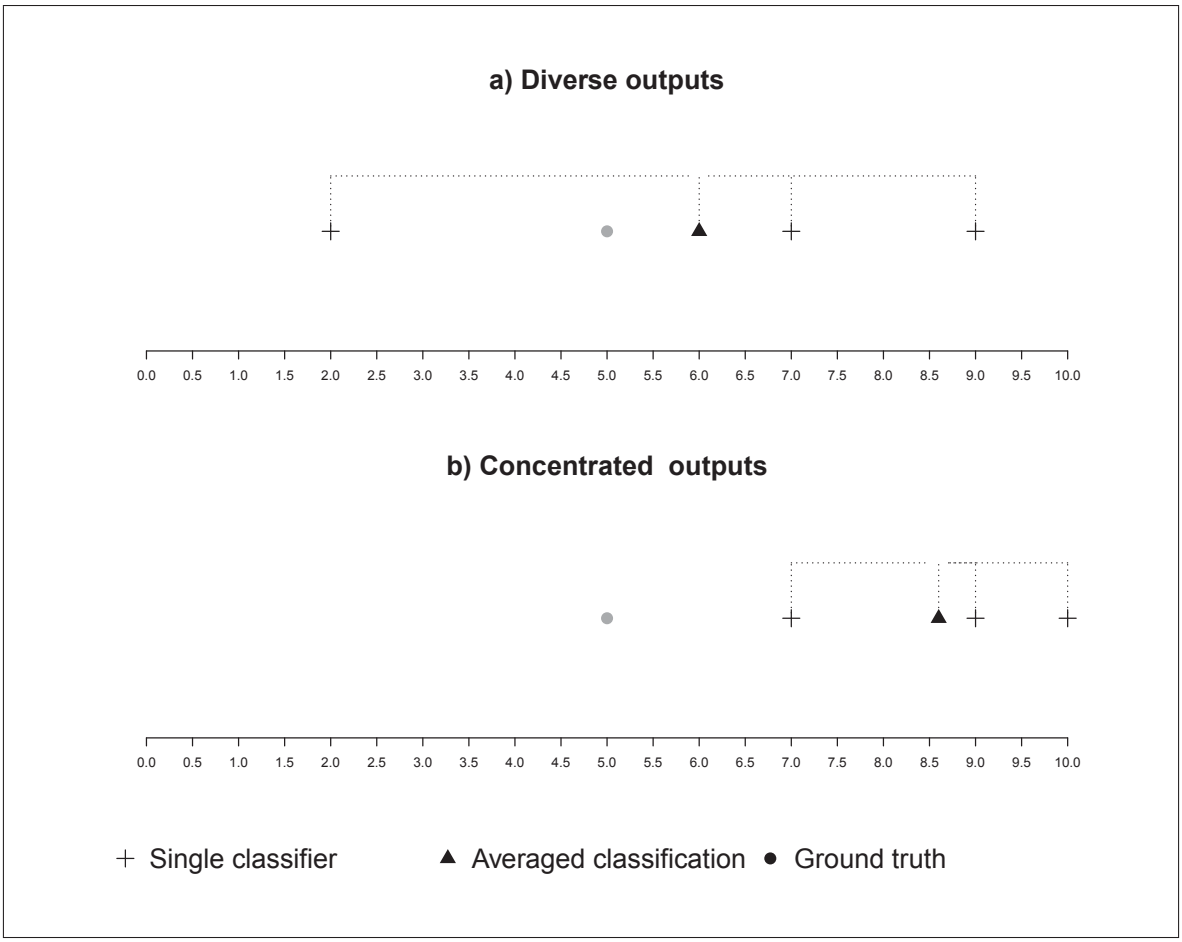


Figure 1.4 Accuracy - diversity trade-offs. Black crosses represent a single classifier output. Black triangle represents the averaged result from 3 single classifiers represented as black crosses. The true output is represented as a gray circle.

the set of observations(Zimek *et al.*, 2013), randomized methods (Liu *et al.*, 2012), by tuning differently the method's parameters (Breunig *et al.*, 2000; Gao & Tan, 2006) and finally with the use of different types of algorithms (Kriegel *et al.*, 2011; Nguyen *et al.*, 2010).

Zimek *et al.* (2013) argues that the same data under analysis is indeed a sample of the true but unknown density distribution; then, building an ensemble based on different samples of the data can provide a better approximation to the true underlying density distribution. The authors in (Zimek *et al.*, 2013) propose an ensemble approach that induces diversity by feeding, in a series of iterations, an outlier detection algorithm (LOF) with different samples of data, this diversity in turn is reflected in an improved detection rate, additionally this mechanism

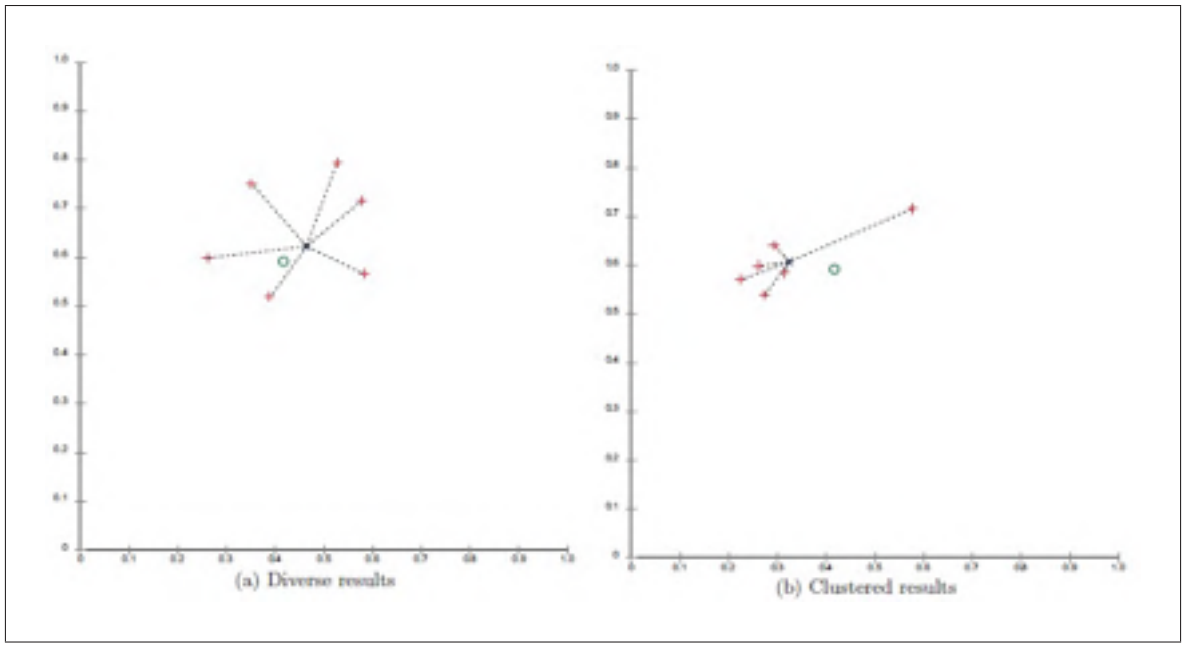


Figure 1.5 Accuracy - diversity trade-off.
Image reproduced from (Zimek *et al.*, 2014).

to induce diversity (samples of data) reduces the overall execution time of the ensemble. In this paper it is argued that each output of the ensemble is superior to a single execution of the base algorithm on the base data; however, in (Aggarwal & Sathe, 2015) it is argued that working in a reduced set of data while keeping the parameter settings fixed (in this case k, as both studies use as base detector nearest-neighbor techniques) can produce biased ensemble components that can or cannot produce an improved detection rate when combined, then the authors in (Aggarwal & Sathe, 2015) proposes to use instead of a relative subsample size a fixed subsample size from 50 to 1000, having then a linear execution time instead of the $O(n^2)$ of the base method.

1.2.3 Bias-variance trade-off

The detection rate of an outlier detector can be affected by different factors, like sample size, algorithms used, parameterization. Thus, the expected error of an outlier detector can be decomposed into irreducible and reducible error (Figure 1.6). The former refers to the limited set

of information for the analysis, the data under observation, in most real-world cases, is only a sample of the true, however unknown data. The latter is characterized as a bias-variance trade-off, which is dependent on different randomization in the data or the algorithms. Determining the sweet spot between bias and variance is an important task in any classification algorithm, this is even more difficult in outlier detection where it is not possible to use ground truth data to find this sweet spot. Variance can be understood as the extent to which the model adapts to the variations in the data, if changing the data with which the model is fitted how much the model will vary. If the model fits the data perfectly then its bias term is zero, and if the model is completely independent of the data the variance term will tend to zero even if the data with which the model is fitted changes. An optimal trade-off of bias and variance will produce a low generalization error (Figure 1.7), this is a balance of model simplicity and complexity.

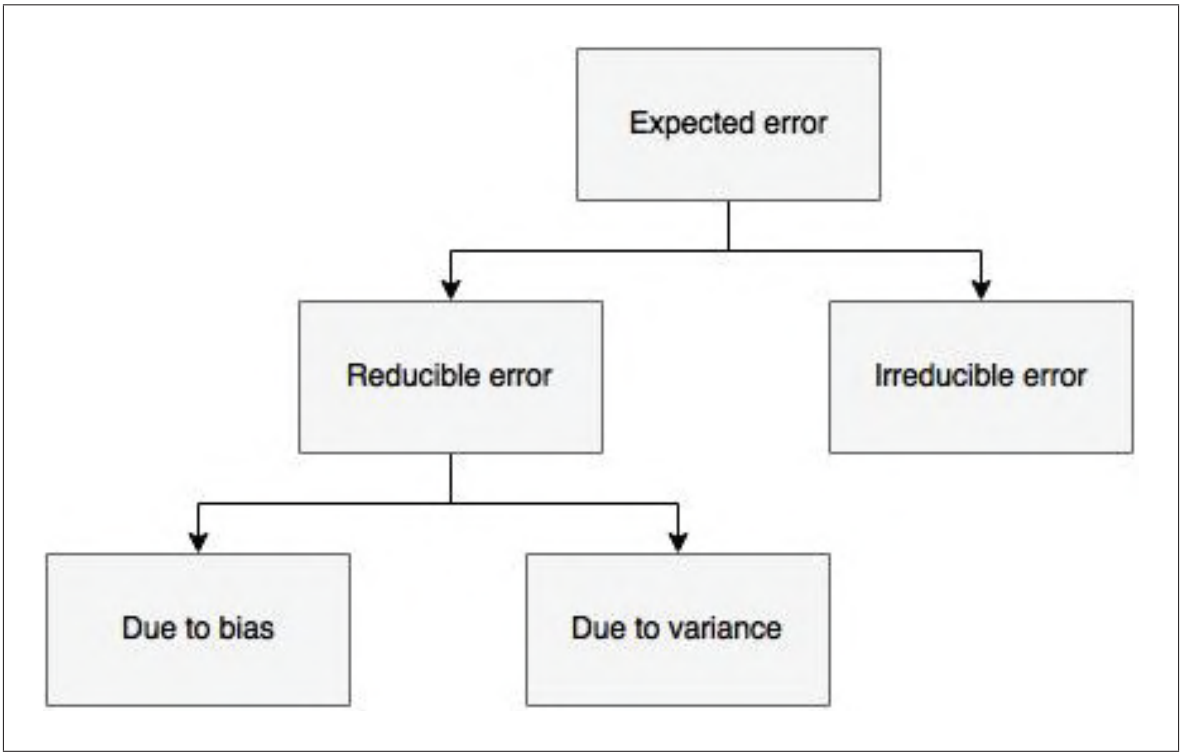


Figure 1.6 Sources of expected error.

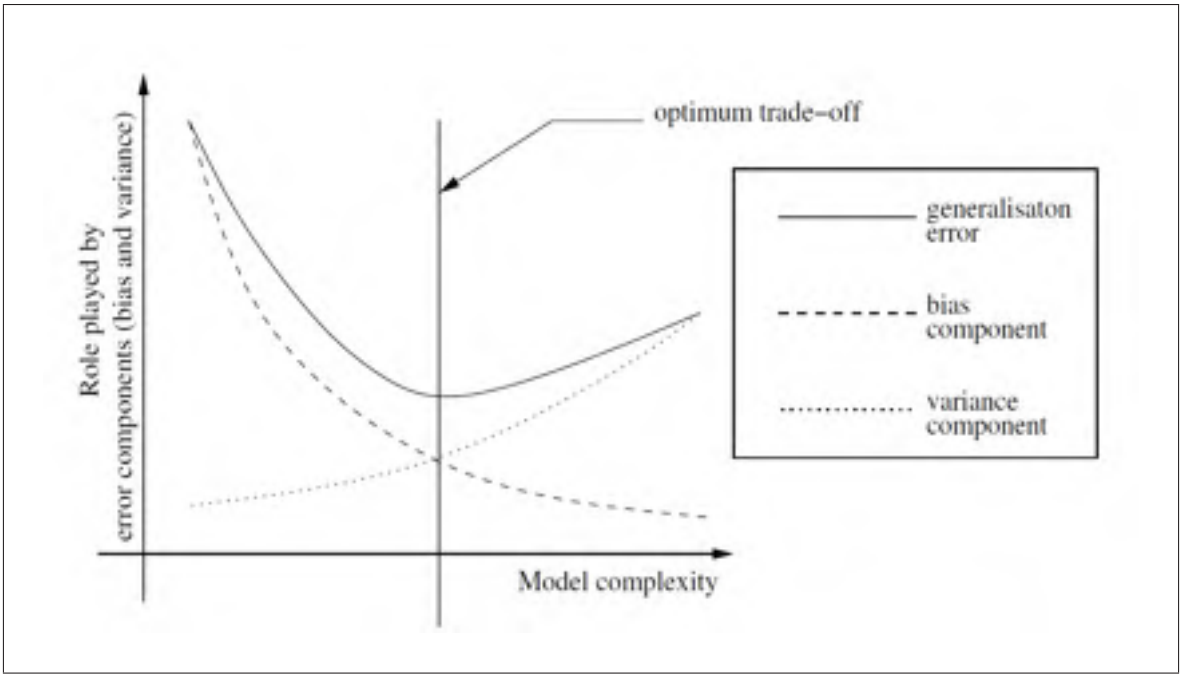


Figure 1.7 Bias and variance Vs. the model complexity
Image reproduced from (Chandra *et al.*, 2006).

1.2.4 Ensembles for unsupervised outlier detection

Ensembles approaches have been developed with the aim of improving the detection rate of a single learner. Their general process is depicted in Figure 1.8. Charu C. Aggarwal defines an ensemble as "any approach which combines the results of either dependent or independent execution of data mining algorithms" (Aggarwal, 2013a). An ensemble has also been proved to generalize better than a single learner (Brown *et al.*, 2005; Tumer & Ghosh, 1996; Brown *et al.*, 2005) .

The literature of ensemble approaches in the classification literature is widely developed with different approaches proposed as bagging and boosting (Breiman, 1996; Freund & Schapire, 1995; James *et al.*, 2015); however, in the outlier detection scenario the quantity of available ensemble approaches is by far more limited. Besides, some unsupervised ensembles for unsupervised outlier detection are not explicitly recognized as such, as they ensembles capabilities are intrinsic to the algorithm (Aggarwal, 2013a). A seminal paper for ensembles of

outlier detectors was proposed by Lazarevic (Lazarevic & Kumar, 2005) where an approach was first categorized as an ensemble of outlier detectors. However, the ensemble idea was already present in the literature but was hidden inside the procedure of single (apparently) outlier detection algorithm. Then, one of the main contributions of Lazarevic was to clearly state the use of an ensemble approach.

Ensembles approaches for unsupervised outlier detection:

- Subsets of dimensions (Keller *et al.*, 2012; Müller *et al.*, 2011)
- Samples of data (Lazarevic & Kumar, 2005; He *et al.*, 2005; Gao & Tan, 2006)

Usually the final output of an outlier detector should be conditioned on a threshold to determine which observations are declared as outliers, any observation above the threshold is declared as outliers, while the remaining points below the threshold are marked as inliers. Lowering the threshold will permit to detect more outliers, true positives, whereas increasing the threshold will miss some outliers, false positives. These threshold can be adjusted depending on the weight given to true positives and false negatives, but usually the aim is in detecting the outlier observations. In medical diagnosis, for example, it is far more important to detect those minority patients with positive results even if this implies having more false positives.

The use of ensembles has a mechanism to improve the performance of a single algorithm has strong bases on the ensemble learning field (Brown, 2011).The field of outlier ensembles is far less explored than that in classification, mainly due to the inherent problems of outlier detection. First, the unsupervised scenario does not allow to have intermediate steps to evaluate the algorithms of the ensemble, like in boosting, and then take further actions based on the evaluation. Second, the unbalanced distribution of outliers and inliers is dramatically high, usually with a proportion of outliers below .05, this smallest number of outliers makes it difficult to use off-the-shelf ensemble classifiers not optimized to detect this minority of points. Third, the absence of class labels makes it impossible to use the common classification path of training the model on training data to posteriorly evaluate it on test data. Even evaluating the results of

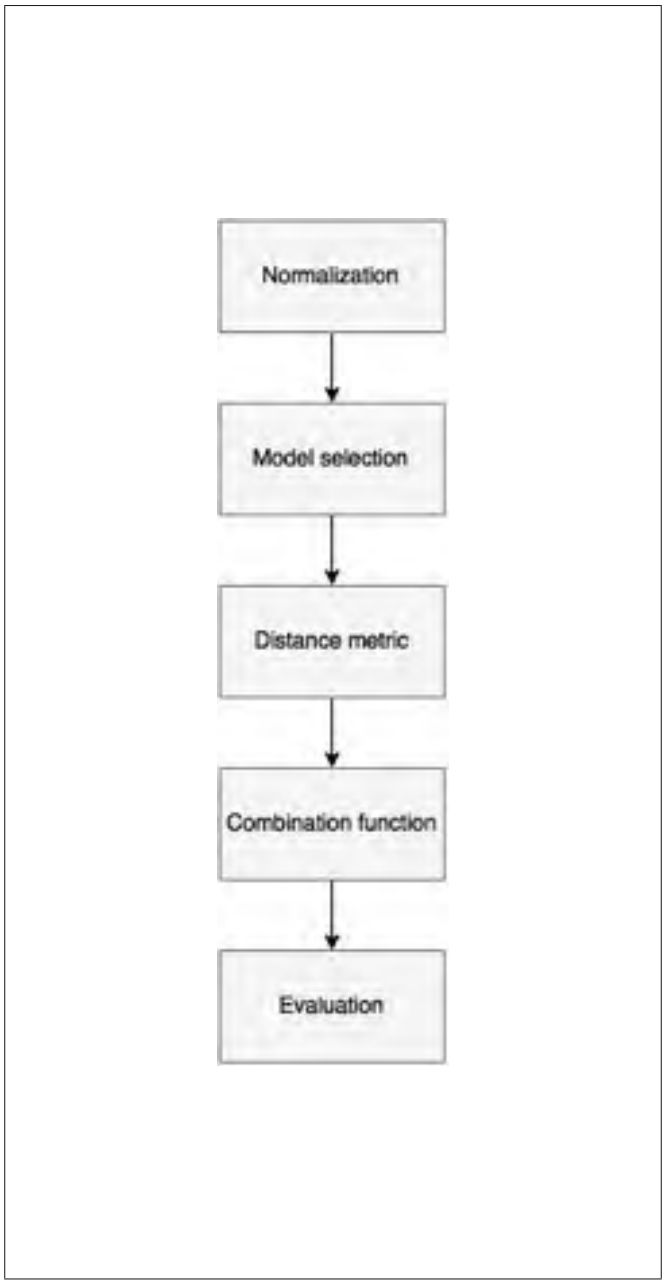


Figure 1.8 Generic ensemble process.

an unsupervised ensemble for outlier detection cannot be done with simple statistical measures like accuracy, this mainly due to the highly imbalanced data in which a simplistic classifier assigning all the observations to the inlier class could achieve a high but misleading accuracy, besides in outlier detection much more weight should be given to the true outliers which is indeed the information we are interesting in. A common metric used to evaluate an outlier

detection algorithm is the ROC curve, which is based on the trade-off of True positive rate and False positive rate, this metric allows to have a better understanding of the performance of a single outlier detection algorithm or in this case an ensemble of these. Another problem is that outliers are usually identifiable only in a subset of the available dimensions of actual real world high dimensionality scenarios.

Classification of ensemble approaches

The field of ensembles for unsupervised outlier detection has been categorized using notions from the classification field where ensembles are classified into three main types depending on the hypothesis space used for learning (Brown *et al.*, 2005): class A by varying the initial conditions with which the learner starts, class B by manipulating the search space and class C by using different weights. In a similar way Aggarwal (2013a) proposes a classification for the unsupervised ensemble scenario by component independence and by component type.

There are different surveys in outlier detection (Aggarwal, 2013a; Zimek *et al.*, 2012; Hodge & Austin, 2004; Patcha & Park, 2007; Chandola *et al.*, 2009). However, they are focused mainly in single algorithms for outlier detection, the survey of Zimek *et al.* (2012) provides a good reference for outlier detection in high-dimensional data but focused on numerical data and using Euclidean distance only.

An ensemble approach can be categorized depending on its component independence or by its constituent components (Aggarwal, 2013a) (Figure 1.9).

Component independence. An ensemble approach is based either on the combination of results from independent executions of an algorithm (another possibility is a set of different types of algorithms) or on a sequence of execution in which previous iterations of the ensemble influence the behavior of the next component in the ensemble. The former type is known as an "independent ensemble" (Aggarwal, 2013a), which is in fact the most prevalent type in the ensemble outlier literature, a classical example is feature bagging (Lazarevic & Kumar, 2005), this type of ensemble characterizes by its ability to deal with the uncertainty found in outlier

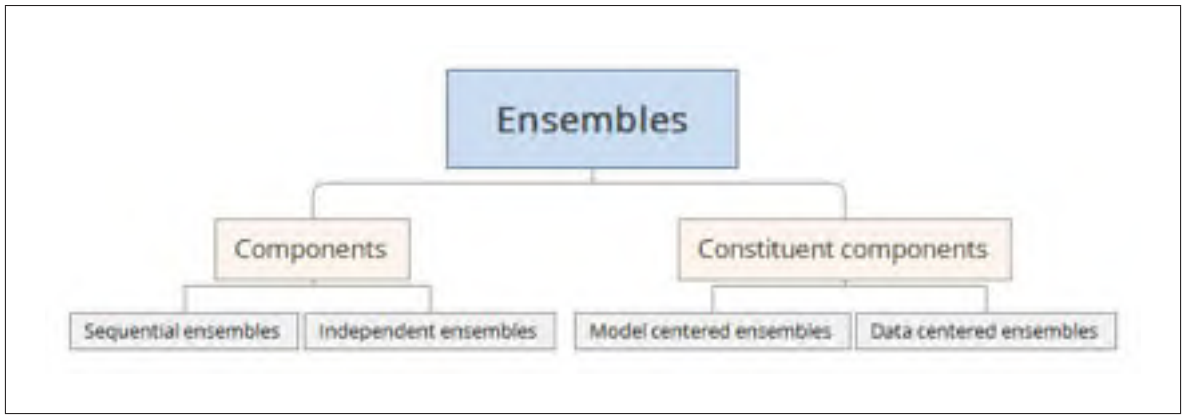


Figure 1.9 Classification of ensembles for outlier detection.

detection, like absence of ground truth and outlier behavior hidden in subspaces, with which it is not possible to use an infallible evaluation to measure neither accuracy nor diversity that allows the selection or assignment of specific weights to different ensemble components; Instead of attempting to construct an ensemble with superb components, and independent ensemble implicitly acknowledges the inherent problems in outlier detection by basing its final result on diverse and independent hypotheses about the outlier behavior of each observation in the data. The latter type is known as "sequential ensembles" (Aggarwal, 2013a) and it is characterized by the sequence or dependence in which the components depend, a classical example, although in the classification literature, is the boosting algorithm (Freund & Schapire, 1995); the advantage of this approach is that if internal evaluation measures are able to produce good results, then it is indeed possible to assign weights depending on the accuracy of the components or instead select only the most accurate components.

Constituent components. An essential component in an ensemble of outlier detector diversity in the results produced by its component members. This diversity can be achieved in two ways: perturbations in the data variations in the based algorithms. The former, also known as data-centered, iteratively feeds the ensemble with different samples of data (Zimek *et al.*, 2013), distinct subset of dimensions (Lazarevic & Kumar, 2005) or both (Pasillas-Díaz & Ratté, 2016a). The latter attempts to induce diversity in the ensemble by simply varying the parameters of the same base algorithm (Papadimitriou *et al.*, 2003) or by using different types of

detectors (Pasillas-Díaz & Ratté, 2016b). Gao & Tan (2006) argues that scores that are by nature incomparable must be brought to a comparable format before combination.

1.3 Parameterization in outlier detection

Outlier detection faces numerous challenges, like the detection of distinct types of anomalies found in the same dataset, absence of class labels, highly unbalanced data, outliers hidden in lower dimensional subspaces, etc. In addition to these challenges an outlier detector, either a single or an ensemble approach, can exhibit a distinct behavior depending on the interaction between the selected parameters of the algorithm and the data.

1.3.1 Interaction algorithm - parameters - data

There is an almost prevalent set of experiments that are usually performed in the approaches for unsupervised ensemble outlier detection present in the literature. In the most general case the researchers examine the interactions of their approach with different data sizes, dimensionalities and some parameter variations. One of the seminal works, feature bagging (Lazarevic & Kumar, 2005), is evaluated using only static synthetic and real-world datasets. A subsampling approach (Zimek *et al.*, 2013) use a more complex set of experiments by considering not only static synthetic and real-world data, but also different sizes of data, sample fractions and ensemble sizes.

1.3.2 Evaluation methods

Beyond outlier detection, the evaluation of the output of any classification algorithm is a crucial step to measure the ability of an algorithm to model the dataset under analysis. However, in the unsupervised scenario this evaluation is tricky and usually there are two main approaches: external measures (Emmott *et al.*, 2013) and internal measures (Marques *et al.*, 2015); the former refers to the use of ground truth labels to evaluate the performance of an algorithm, obviously this step cannot be done in real-world scenarios, as the word “unsupervised” clearly

states the inexistence of labels; however, this approach is often used in the literature to evaluate proposed outlier detection algorithms, using datasets from the classification field and adapting them to the outlier detection scenario by holding the true labels until the evaluation phase; this clearly is not the best way to evaluation an outlier detection algorithm as the class used as the inliers class can also have true outliers originated directly in the application domain, then this adaptation is measuring only the ability of the algorithm to detect the minority class selected by the user. The latter refers to an evaluation based on whether or not the algorithm output fit certain assumptions relative to what is a good clustering, density formation, etc. Also algorithms like SELECT (Rayana & Akoglu, 2016) produce its own internal measure by estimating a "pseudo ground-truth" and then using this artificially created labels to decide which ensemble members to drop from the final output, this decision is based on the capacity of each algorithm to improve the accuracy of the ensemble.

ROC curves (Figure 1.10) take into account the imbalanced scenario of outlier detection, which makes them particularly useful for outlier detection. ROC curves endpoints are invariably (0,0) and (100,100). A random classification will be represented as a curve near the diagonal, with an AUC around 0.5. A perfect curve has a vertical line on X axis (false positive rate) and a vertical line at 1 on the y axis (true positive rate), this indicates a perfect classification, and where at a moving threshold t all the outliers are ranked higher than inliers, the AUC for a perfect classifier is 1. Being based on false positive rate and true positive rate ROC curves are a good fit for imbalanced scenarios, like is the case of outlier detection.

ROC and ROC AUC are widely used to measure outlier detectors performance, it has been argued (Schubert *et al.*, 2012) that a disadvantage of ROC analysis is that while it certainly captures the relative rank of each outlier scores it fails to take into account the information contain in the scores; then, the authors in Schubert *et al.* (2012) proposed to use beside a ROC analysis a ranking similarity measure that can provide further hints about the diversity of ensemble components. Despite the appealing characteristics of the ranking similarity measure to improve ensemble diversity, the main approach to measure the results of outlier detection algorithms continue to be based on ROC analysis, this can be due, mainly, to the ease with

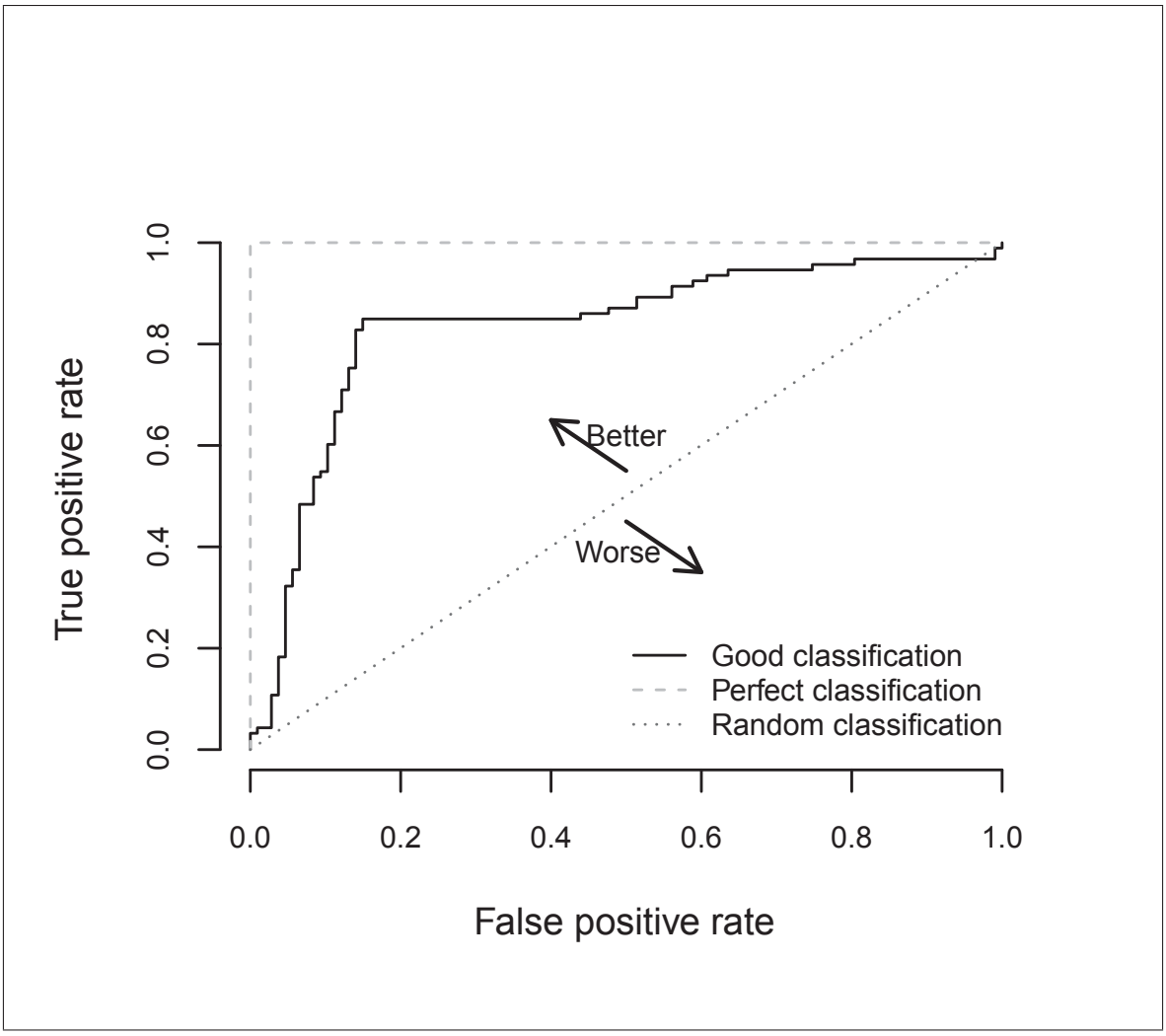


Figure 1.10 ROC curves for a perfect, good and random classification. Upper arrow indicates the direction in which the classification is better than random, lower arrow signals a classification worse than random.

which is possible to compare multiple results in a single ROC curve graph and a table with AUC.

Another type of measure is to evaluate the results of an outlier detection algorithm is to use the top n results, which is known as precision at k (Craswell, 2009). In this setting only the top k outliers are subject to evaluation, as they are the outputs that the algorithm classified with the higher probability of being outliers; however, this approach requires to know an extra parameter k, which is completely domain and data dependent; using precision at a threshold

k the detection of outliers just below the threshold is punished (Schubert *et al.*, 2012). Let's assume a scenario with 2 outliers and an outlier detection algorithm ranks the true outliers in the positions 9-10, but *k* is set to 2, meaning that it will expect to find the outliers in the top 2 positions, in this case precision@k will be equal to zero, whereas setting *k*=10 will return a precision@k=0.2. Overall, there are three main problems with precision@k, first it doesn't take into account the relative position of the outliers in the rank, being the same it the algorithm ranks them very high or low while they are below the threshold *k*, second being unsupervised outlier detection a field whose of its main characteristics is the absence of information about the dataset like ground truth, let alone to know how many outliers are expected to be found in the data, setting *k* higher than the number of true outliers will yield imperfect results even for an algorithm that rank perfectly the outliers. Then, despite that this measure is still used in fields like information retrieval, in outlier detection it's used its limited to have, at least, an estimate about the number of outliers in the dataset.

The trade-off between outliers and inliers can also be measured with the use of precision and recall, the former measures the percentage of detected outliers which are truly outliers, the latter refers to the percentage of truth outliers which were actually classified as outliers. A precision-recall curve can be used to visualize the trade-off between these two measures.

1.4 Current limitations

In this section we accentuate the limitations of current state of the art approaches for outlier detection, aiming toward the detection of hidden and diverse outliers.

1.4.1 Limitation 1. Inadequacy of an outlier detector to identify different types of outliers

Current iconic algorithms for outlier detection are highly specialized towards a specific type of data. However, such specialization is also accompanied by blindness to distinct types of outliers. Being outlier detection, at least in the most interesting and difficult cases, an unsupervised process with limited or even inexistent information about the data under study, the selection

of a single detector is if not infeasible, at least flawed. Even ensemble approaches for outlier detection exhibit such overlooking behavior, being, in general optimized towards increasing detection rate of a specific type of algorithm; a blind adaptation of such ensemble approaches to operate with different types of algorithms is a complicated process as distinct types of detectors provide outputs which are not directly comparable. Moreover, without further knowledge about the data, an external evaluation of each ensemble component is inconceivable, deriving in inability to assign weights depending on the performance of a component on a specific type of data. Therefore it is important to devise an approach to combine distinct algorithms for outlier detection while devising a mechanism to combine and individually weight each detector depending on an internal and unsupervised evaluation of its ability to detect outliers in a specific set of data.

1.4.2 Limitation 2. Lack of computationally inexpensive approaches focused in the detection of outliers hidden in lower dimensional spaces

An ensemble of classifiers, beyond the outlier detection scenario, is built on top of two fundamental concepts: accuracy and diversity. Even in the supervised scenario these concepts are not fully understood in the context of an ensemble setting and consequently there isn't a strong theory explaining how diversity affects the accuracy of an ensemble; moreover, without fully comprehending diversity the task of designing a diverse and accurate classifier is complicated, as Brown *et al.* (2005) clearly stated, "It seems the amorphous concept of diversity is elusive indeed". While, in a supervised setting, it is possible to measure the accuracy of a single classifier in the presence of class labels and also obtain a proxy to diversity by measuring the disagreement between classifiers, also known as diversity in errors, in outlier detection the unsupervised nature of the outliers makes the task even more complicated, without class labels it is not possible to use mechanism to explicitly induce diversity in the ensemble, instead implicit methods have to be used to induce diversity by perturbing the samples of data, dimensionality or the algorithm's parameter settings.

Simple approaches for outlier detection explore the data by searching in the whole set of dimensions to find outliers whose behavior is present only on the combination of all the available attributes. This approach to outlier detection offers a limited insight to the final user as it is only able to detect those uninteresting outliers, moreover, these approaches are deemed by the sparsity of points in high dimensionality, where each observation is seen as an outlier. However, the exploration of all possible subspaces to find the specific combination of attributes where an interesting outlier is located is an infeasible task due to exponential increase in processing time as the number of dimensions to analyze increases. Thus, a challenging task in outlier detection is the identification of outliers hidden in lower dimensionalities of the data while maintaining a low execution time.

1.4.3 Limitation 3. Absence of a comprehensive study of the interaction parameter setting - dataset - outlier detection algorithm

There are a few studies in the literature studying the behavior of an outlier detection algorithm when interacting with distinct combinations of parameter settings and data scenarios. These studies are oriented to the effects of bias and variance (Aggarwal & Sathe, 2015) , combination measures (Schubert *et al.*, 2012), normalization functions (Kriegel *et al.*, 2011), parameter settings(Campos *et al.*, 2015), attributes and/or subsample variations (Zimek *et al.*, 2013; Pasillas-Díaz & Ratté, 2016a; Lazarevic & Kumar, 2005), combination of different types of algorithms (Nguyen *et al.*, 2010) and evaluation measures (Campos *et al.*, 2015). Despite that most of the recent advancements in outlier detection are essentially oriented towards similarity based learning, either in the form of a single algorithm or an ensemble of these, there is a gap in the study of the interaction between distance metric - dataset - detector. Accordingly, all the approaches proposed in the literature are evaluated using in most of the cases a single distance metric, overseeing the impact on the detection rate and processing time that different distance measures can have when interacting with a specific dataset.

CHAPTER 2

AN UNSUPERVISED APPROACH FOR COMBINING SCORES OF OUTLIER DETECTION TECHNIQUES, BASED ON SIMILARITY MEASURES

José Pasillas[1], Sylvie Ratté[1]

[1] Département de génie logiciel et des technologies de l'information , École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Paper published in the journal « Electronic Notes in Theoretical Computer Science » from Elsevier, August 2016.

ABSTRACT

Outlier detection, the discovery of observations that deviates from normal behavior, has become crucial in many application domains. Numerous and diverse algorithms have been proposed to detect them. These algorithms identify outliers using precise definitions of the concept of outliers, thus their performance depends largely on the context of application. The construction of ensembles has been proposed as a solution to increase the individual capacity of each algorithm. However, the unsupervised scenario (absence of class labels) in the domains where outlier detection operates restricts the use of approaches relying on the existence of labels. In this paper, two novel unsupervised approaches using ensembles of heterogeneous types of detectors are proposed. Both approaches construct the ensemble using solely the results produced by each algorithm, identifying and giving more weight to the most suitable techniques depending on the particular dataset under examination. Through experimental evaluation in real world datasets, we demonstrate that our proposed algorithm provides a significant improvement over the base algorithms and even over existing approaches for ensemble outlier detection.

2.1 Introduction

Our capacity to collect and store data increases in an exponential manner but our capacity to analyze it has not followed the same trend. Despite the explosion of available data, the discov-

ery of truly interesting patterns is a rare event. Outlier detection the discovery of observations that deviates from normal behavior has been widely studied in recent years (Patcha & Park, 2007; Hodge & Austin, 2004; Chandola *et al.*, 2009), resulting in a set algorithms designed to detect these rare but potentially crucial events. In some specific contexts an outlier is a data point that can be considered either as an abnormality or noise, whereas anomaly refers to a special kind of outlier which is of interest to the analyst. However, the terms outlier and anomaly, in general, have been used interchangeably in the literature (Chandola *et al.*, 2009).

One of the core definitions of outliers was made in 1980 by Grubbs (Grubbs, 1969): "An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs". However, this definition lacks one important characteristic, this is, the case where the outlying points conglomerates to form their own group of outliers; Barnett and Lewis (Barnett & Lewis, 1994) improved the definition of outliers by considering as outlier not only a single and isolated point, but also a group of points deviating from the normal behavior.

The effect of undetected outliers in different application domains (i.e. medical, intrusion detection, fraud detection, geographical) could have deep and disastrous consequences. An example is the detection of breast cancer where an undetected positive case implies an untreated patient; another example is a failed attempt to detect strange behavior in the use of a stolen credit card resulting in a financial impact for the credit card holder. In both of these examples, the minority of the cases represents the class of interest.

The process of outlier detection represents a very specific classification scenario: first, the quantity of outliers is very small in proportion to the quantity of normal instances; and second, the use of labels (supervised approach) in outlier detection is limited due to the fact that, by definition, the outliers that we are trying to detect represent a new or unseen behavior. Despite the fact that some algorithms (techniques) can operate using only labels for the normal class (Noto *et al.*, 2010) (semi supervised approach) and use this information to increase the detection rate, unsupervised approaches have the undeniable advantage of operating over unlabeled

data. Furthermore, unlabeled data are usually easier to obtain and represents the more common scenario in outlier detection (Eskin *et al.*, 2002).

The use of an unsupervised outlier detection approach also has the benefit of avoiding the bias introduced by training an algorithm with anomalous observations, labeled wrongly as normal data, causing the misclassification of future similar observations.

Due to the large spectrum of domains where outlier detection can operate, there are a wide variety of outlier detection algorithms mainly based on: classification, clustering, nearest neighborhood and statistical approaches (Chandola *et al.*, 2009). However, their use is application dependent; no single outlier detection algorithm is best suited for all the different data scenarios that we could encounter in real world datasets (Lazarevic *et al.*, 2003). Some algorithms work better when the data tend to form clusters, whereas others are most suitable to use in the presence of neighborhoods in the data.

Despite the fact that by working on an unsupervised scenario it is not possible to know which algorithm is better for a specific dataset in advance, the performance of these algorithms can be improved.

Similar to ensemble classifier learning, where heterogeneous assumptions are used to produce a unified output (Oza & Tumer, 2008; Opitz & Maclin, 1999), in ensemble outlier detection, diverse (heterogeneous) assumptions are also needed to produce a meaningful result, potentially complementing each other. There is no gain if the techniques that form the ensemble produce exactly the same output.

The more common scenario is to construct a diversified ensemble with techniques whose results are uncorrelated, using class labels (supervised approach) and algorithm outputs to determine the similarity between techniques. However, when the class labels necessary to compare the agreements between techniques are absent (as is the case in an unsupervised setup), a different way to establish diversity must be found. In this regard, some approaches ensure diversity in the ensemble by providing different samples of features, but apart from the fact that mul-

tiple iterations are required to analyze each sample, some datasets will require the use of the complete set of features to identify the outlier observations.

The approach we are proposing reaches diversity not by comparing the output of the algorithms and the class labels, but by creating the ensemble with a varied set of algorithms.

Combining outputs of different classifiers is not a novel task; however, outlier detection has to face two additional problems (Lazarevic & Kumar, 2005). First, an ensemble of classifiers works with discrete labels whereas outlier detection is mainly concerned with scores. Second, an ensemble of classifiers generally relies on the existence of training data (supervised approach), whereas outlier detection generally does not have access to labeled data (unsupervised approach).

We propose two novel approaches based on a weighted combination of outlier detection algorithms, both of which give more weight to algorithms whose outputs offer an expected better performance for a specific data representation, and improve the differentiation between outlier and inlier by increasing the relative distances between the scores of outliers and those of inliers.

The rest of the paper is organized as follows: section 2.2 provides a background about techniques for outlier detection, ensemble methods, and evaluation procedures; section 2.3 introduces our approach in detail; section 2.4 illustrates some experiments with real life datasets and section 2.5 concludes our research and discusses the scope for future work.

2.2 Background and related work

Outlier detection is a very active research area where new approaches are proposed each year. Nevertheless, the detection of outliers was first contemplated in the statistical community in 1887 (Edgeworth, 1887). Since then, different techniques based on various approaches such as classification, clustering, density-based and statistical inference have been proposed.

An important characteristic of an outlier algorithm is its output, which can be either a score or binary label (Aggarwal, 2013b). The former type of output assigns a score to each observation

and in general can be used to rank the observations depending on its level of outlierness. The latter assigns binary labels, commonly using 1 for outliers and 0 to designate normal observations (inliers).

A score has the advantage of retaining more detail by providing a degree of outlierness, whereas a binary output offers a more simplistic classification of an observation as either inlier or outlier. Despite the convenience of a binary output, the information retained in the scores could offer more insights about the outlierness of an observation.

The construction of an ensemble of outlier algorithms seems like a viable solution when the objective is to increase the detection rate of outliers (e.g. breast cancer detection) while diminishing the variance introduced by each outlier detection algorithm. However, no gain will be obtained by using algorithms whose results are identical. Therefore, two important factors must be taken into account when constructing an ensemble: accuracy and diversity. Accuracy measures the output quality of each algorithm, while diversity endeavors to build an ensemble whose results are distinct and, in theory, complementary. Accuracy depends on the right association of technique and dataset; diversity can be established using variations of the search space (data and feature sampling) or by the use of different types of algorithms (Tan & Maxion, 2005). Combining different types of algorithms could yield better performance than simply using parametric variations of the same algorithm (Schubert *et al.*, 2012). However, a balance between accuracy and diversity is needed in order to obtain an improved ensemble detection rate (Zimek *et al.*, 2014); highly diverse, but inaccurate algorithms, results in an ensemble whose components are truly diverse, but without the accuracy component is unable to converge near the true classification output, resulting in an ensemble whose detection rate is below that of its individual members.

The process of building an ensemble involves three main considerations: the choice of the algorithms, the organization (modular or ensemble) and the combination method (Canuto *et al.*, 2007). A multiclassifier can be categorized as modular or ensemble. A multiclassifier is modular when each member is responsible for a specific part of the process and the algorithms

are used in a series of steps, using the results of the previous algorithm. It is an ensemble when each single member works on the same search space and a combinatorial process joins the results to produce a unified output. In this paper we are focusing on the latter type. The most important component is the combinatorial approach chosen so that each single member (classifier) contributes to improve the overall performance.

One critical factor in the construction of an ensemble is to mix members (algorithms) whose errors are not identical; doing so assures us that these members complement each other, therefore producing potential improved results. However, the majority of such approaches assume that a measure of accuracy for each member is available, using class labels for each observation. Still, considering that outlier detection is mainly an unsupervised field, it is not practical to measure accuracy using output labels. In our proposed approach, we do not assume highly accurate classifiers trained with the use of labeled data; instead we estimate accuracy by considering only the output scores of each algorithm and attempting to achieve diversity using different types of outlier detection techniques.

In our empirical studies, four detectors are used: a density-based approach (Local Outlier Factor or LOF), two distance based approaches (k-means & hierarchical clustering) and a statistical based approach (modified boxplot). The density-based approach LOF is considered one of the most performing outlier detection algorithms (Lazarevic *et al.*, 2003). This technique computes a degree of isolation that depends on two factors: first, the distance between a point and its neighbors, and second, the density of the neighborhood. The detection of outliers using boxplots (Torgo, 2010; Laurikkala *et al.*, 2000) is one of the most simple model based techniques; this statistical approach makes no specific assumptions about the data distribution determining as outliers those points beyond a specific threshold. The first distance based approach relies on the k-means algorithm (Hartigan & Wong, 1979); the data is divided into different groups depending on the closest centroid; the outlierness of a point is equal to the distance to its closest centroid. An outlier algorithm using hierarchical clustering (Torgo, 2007) divide the data into binary clusters recursively until the data cannot be divided any further; in this case outliers consist of those observations that present more resistance to being merged into a cluster.

While increasing the detection rate of the ensemble using a combination of only highly accurate classifiers seems like a good idea, the unsupervised nature of the datasets where outlier detection operates is a limiting factor. When considering an unsupervised scenario, it is crucial to use selfsufficient measurements of diversity that are based only on the results of the members of the ensemble and not assume the existence of labels for the normal instances (semi-supervised approach) or labels for both normal and outlier instances (supervised approach).

Therefore, our focus is on self-sufficient measurement of diversity. Previous studies such as feature bagging (FB) (Lazarevic & Kumar, 2005) use variations of the search space to induce diversity in the ensemble; a similar study (Nguyen *et al.*, 2010) uses both variation in the search space and different outlier detection techniques.

The feature bagging approach starts by randomly choosing without replacements different subsamples of features; then in a series of rounds, each outlier technique analyzes these subsamples producing a set of output scores. Finally, the process of joining the scores can be performed with any of the two methods provided by the authors of feature bagging: *Breadth First* and *Cumulative Sum*.

The *Breadth First* method first sorts the outlier scores from all the iterations of feature bagging, next takes the index of the record with the highest score and then inserts its index in a vector, and so on. If an index is already in the vector, it is omitted. The final output is a vector of indices pointing to its corresponding scores.

The second variant of feature bagging is *Cumulative Sum*. This method simply adds up the scores of each iteration of feature bagging, and the outliers are those observations with a resulting high score.

The *Breadth First* approach is exposed to a critical observation: it is highly sensitive to the order in which the outlier detection algorithms were applied. This means that the first technique in the ensemble has priority to decide about the outlierness of a given data record. Also, the methodology of this approach does not indicate how to establish the order of the algorithms.

Cumulative Sum reports better performance overall when compared with the *Breadth First* method (Lazarevic & Kumar, 2005). This way of combining the outputs overcomes the order problem of the the members in *Breadth First*. However, neither of the two variants of feature bagging takes the use of different types of algorithms into account.

The authors of feature bagging used only one algorithm (LOF) for their experiments and there is no mention on how to join scores in different scales. To achieve better performance, their experiments assume the existence of labels for the normal instances (inliers).

The authors of feature bagging (Lazarevic & Kumar, 2005) report improvements on performance over a single outlier detection technique; their results provide solid foundation upon which to compare new approaches. However, we hypothesize that better performance can be achieved by joining the outputs of different types of algorithms and setting specific weights, without assuming any knowledge of the output labels.

Receiver operating characteristics (ROC) curves are very useful when measuring the performance of outlier detectors. These curves consist in plotting the true positive rate (TPR=ratio of true positives to actual positives) versus the false positive rate (FPR=ratio of false positives to actual negatives) using a variation of a discriminant threshold. For that matter, the area under the curve (AUC) is often used as the benchmark in outlier analysis (Lazarevic *et al.*, 2003; Lazarevic & Kumar, 2005; Schubert *et al.*, 2012; Nguyen *et al.*, 2010; Kriegel *et al.*, 2011; Fawcett, 2004). AUC is the probability that a randomly selected positive instance will be ranked higher than a randomly selected negative one. AUC is a convenient metric to evaluate the performance of outliers algorithms when it is not possible to predetermine a threshold and instead of a ROC curve a single measure is required (Bradley, 1997). The higher the AUC, the better the expected performance of the technique; an AUC=1 indicates a perfect performance, whereas an AUC=0.5 indicates performance similar to a simple random choice.

Besides ROC curves and AUC, other commonly used evaluation measures are accuracy and precion@n (Craswell, 2009). The former, is commonly used in the classification scenario to evaluate the results of classification algorithms; however, in outlier detection the highly imbal-

anced datasets can bias this measure; e.g. a simplistic classifier assigning all the observations to the inlier class will produce a high and misleading accuracy value, when truly it is erroneously classifying all the outlier observations, which are in outlier detection the observation that the final user is, indeed, trying to find. The latter, is another measure that can be used to evaluate outlier detection algorithms; however, his measure is highly sensitive to the selection of n (Campos *et al.*, 2015); e.g. in a toy scenario with only 2 outliers and 100 inliers, an outlier detection algorithm ranks the true outliers in the third and fourth position (almost perfectly considering an unsupervised outlier detection scenario), a selection of n=4 would result in a precision@n=0.5; however, setting n=2 would give a precision@n=0, despite that the classifier has indeed highly classified the outliers. Precision@n requires the user to have at least some knowledge about the expected number of outliers in the data; in outlier detection, being in general an unsupervised setting, it is neither possible to know in advance the ground truth class labels nor the number of outliers present in the data.

ROC curves are widely used in the literature to evaluate unsupervised outlier detection algorithms, then their use facilitates the comparability with previous research works (Tan & Maxion, 2005).

2.3 The approaches

We propose two novel approaches for combining the outputs of heterogeneous outlier detection algorithms in an unsupervised scenario: ensemble of detectors with correlated votes (*EDCV*) and ensemble of detectors with variability votes (*EDVV*).

With prior knowledge of which detector will work better for each dataset, it is possible to predetermine a specific weight for each algorithm. However, working in an unsupervised approach requires measuring the ability of each algorithm independently of the existence of labels. The main difference between *EDCV* and *EDVV* is the measure used to estimate the coefficients or weights when the outputs of the algorithms are compared. *EDCV* uses correlation coefficients as a similarity measure, whereas *EDVV* uses the mean of the absolute deviations

between outputs (MAD) as a dissimilarity measure in the form of 1-MAD. The two also use a modified boxplot method to determine the number of outlierness votes that each observation receives from the algorithms. In this way, both approaches assign weights but in two different ways: first, by measuring the performance of each algorithm over the specific dataset (similarity/dissimilarity measures), and second, by giving a number of votes to each individual score produced by each algorithm.

At this point, two different measures (correlation for *EDCV* and MAD for *EDVV*) are used to determine the individual performance of the algorithms over a specific dataset. The similarity/dissimilarity measures assign specific weights to each one of the algorithms of the ensemble, giving more influence to those algorithms whose outputs are similar.

The approaches use two different similarity/dissimilarity measures for numerical values: correlation and MAD; we use them to measure the similitude between the outputs of different classifiers. The former can be used to evaluate the statistical correlation between different outputs; also it is indifferent to the scale of the input values and will produce a result of 1 for perfectly correlated values, 0 for uncorrelated values and -1 for negatively correlated values. The latter is used to measure the absolute deviation between different outputs. MAD produces results relative to the scale of its components. Whereas MAD tends to assign low values to similar scores, correlation coefficient assigns high values to correlated scores.

2.3.1 General approach

The two approaches we are proposing are based on the same procedure described in Algorithm 2.1, however, they differ critically in the way they assign the weights to each algorithm. In this subsection we present the first phase of both approaches leaving the weight assignation for the following subsections 2.3.1.1 (*EDCV*) and 2.3.1.2 (*EDVV*).

As shown in Algorithm 2.1, a given dataset (*DS*) of size m is first examined by applying each of the algorithms in a series of T rounds, where T represents the number of algorithms available in the ensemble. For testing purposes we are using $T = 4$. Nonetheless, T can take differ-

Algorithm 2.1 General Approach for combining outlier detection scores

input : Given a dataset DS=((x_1),(x_2)...(x_m)) of size m, where x_i represent a specific observation. T equals the set of algorithms in the ensemble; T_i refers to a specific algorithm in T.
output: Ensemble outlier scores $Ffinal$

```
1  procedure GENERAL APPROACH()
2      for each i in t ∈ T do
3          Select randomly, without replacements, a set of features F(t) from D of random
             size between d/2 and d-1 ;
4          Apply outlier algorithm T_i to DS;
5          The output of T_i is output score F_i;
6          Standardize F_i;
7      end
8      Determine votes (V);
9      Determine weights (W);
10     Combine the output scores F and produce a final ensemble output Ffinal;
11 end procedure
```

ent values, meaning that our approach is not constrained either to the use of specific outlier algorithms or by the number of them. We expect that our approach can be applied using the majority of outlier detection algorithms that are capable of producing results in the form of scores.

The different algorithms for outlier detection produce scores on different scales; for example while LOF tend to produce values close to 1, hierarchical clustering produces results with a much larger range. We have determined that the best way to normalize these results is to use a standardization procedure. Standardization is frequently used as a normalization method in ensemble outlier detection (Hawkins, 1980; Lazarevic & Kumar, 2005), bringing the different outputs to comparable scale and maintaining the relative larger scores of the outliers compared with those of inliers, avoiding in this way that algorithms with the largest range of results dominate the final result. The standardization method we are using consists in transforming the output scores (F) into Z scores with the conventional procedure *Z= (Xi-mean)/SD* (where *SD* is the standard deviation). This standardization step allows for an observation with a large score in one technique to maintain a large value after joining the ensemble.

Using these standardized outputs (F) from each algorithm, we then apply a modified boxplot technique to detect those outputs whose deviations are greater than the rest. In this way we produce a vector of votes (V) of size $m * T$ (number of observations multiplied by the number of algorithms) that contains the number of votes of each algorithm for each observation. An observation receives a vote if its score is greater that 1.5*IQR (where IQR is the inter quartile range). We determine the IQR in the conventional way (Tukey, 1977) IQR=Q3-Q1, where Q3 and Q1 stand for third quartile and first quartile respectively. Accordingly, the output matrix V in this step has the same dimensions as the matrix containing the standardized scores F. Each score in F will have a corresponding number of votes in V; for example V_{ij} corresponds to the number of votes assigned to F_{ij}.

The following two subsections 2.3.1.1 *EDCV* Approach & 2.3.1.2 *EDVV* Approach) describe the calculation of the matrix of weights (W). Although both approaches used the same general procedure, they differ in how they calculate the matrix W.

The matrix W measures the individual capacity of each algorithm over the specific dataset under examination, increasing the weight received for outliers while maintaining those of inliers. While it is obvious that each outlier algorithm has already assigned an intrinsic weight with the scores assigned to each observation, we attempt increasing the weight of outliers, while maintaining those of inliers, to have a better differentiation between outlier and non-outlier.

The main difference between the votes V and the weights W is that the votes are intended to increase the difference between outliers and non-outliers and are produced individually for each observation whereas the weights will not be specific to a particular observation but instead reflect the apparent capacity of the algorithm over the dataset under examination.

The subsection 2.3.2 explains how the F scores are combined using the votes (V) and weights (W) to produce the final score, *Ffinal*.

Algorithm 2.2 The *EDCV* approach for joining outlier scores

output: Return matrix of weights $W = \{w_1, w_2 \ldots, w_n\}$

procedure `EDCV()`
  Compute matrix (C) of correlation coefficients between the standardized output scores F;
  For each technique, produce w_n as the average of its corresponding column of correlations C_m;
  for *each n in T* **do**
    $$Ofinal = \frac{(\sum_{m=1}^{T} Cmn) - 1}{T - 1}$$
  end
end procedure

2.3.1.1 EDCV approach

The process of obtaining the weights (W) for each algorithm (T) using the *EDCV* approach is displayed in Algorithm 2.2. First, we obtain a matrix of correlations C (2.1) with dimensions m=size of T by n=size of T by calculating the correlation between the standardized scores F. For example, as represented in (2.1), C_{mn} stands for the correlation coefficient between scores F_m and F_n. Next, we divided the average of the correlations corresponding to each F_n by the size of T to obtain the matrix W; given that the correlation of an algorithm with itself is meaningless as it corresponds invariably to a perfect correlation with value 1, then we subtracted 1 from both the numerator and denominator. The resulting matrix of weights W=$\{w_1, w_2 \ldots, w_n\}$ represents the specific weights for each algorithm.

$$C = \begin{array}{c} \\ F_1 \\ F_2 \\ \vdots \\ F_n \end{array} \begin{array}{c} \begin{array}{cccc} F_1 & F_2 & \ldots & F_n \end{array} \\ \begin{pmatrix} C_{11} & C_{12} & \ldots & C_{1n} \\ C_{21} & C_{22} & \ldots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{m1} & C_{m2} & \ldots & C_{mn} \end{pmatrix} \end{array} \tag{2.1}$$

2.3.1.2 EDVV approach

The second variant of our approach, *EDVV*, obtains W with the process displayed in Algorithm 2.3. First, a matrix (D) (2.2) with dimensions m=size of T by n=size of T is produced by calculating the MAD between the standardized scores F.

$$D = \begin{array}{c} \\ F_1 \\ F_2 \\ \vdots \\ D_n \end{array} \begin{array}{c} \begin{array}{cccc} F_1 & F_2 & \dots & F_n \end{array} \\ \begin{pmatrix} D_{11} & D_{12} & \dots & D_{1n} \\ D_{21} & D_{22} & \dots & D_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ D_{m1} & D_{m2} & \dots & D_{mn} \end{pmatrix} \end{array} \tag{2.2}$$

Note that the matrix D is similar in size and structure to that produced by the other variant of our approach *EDCV*; however, in the present case the values of the matrix D (2.2) represent deviations and not correlations. MAD assigns lower values to similar output scores and our general framework expects that the highest weights of W represent the most suitable algorithms, so when feeding the matrix D with MAD values we transform them to a compatible form with our general approach by using the complement 1-MAD.

After this step, the average of the each F_n in matrix D is divided by the size of T-1 to produce the matrix W. This is different from the *EDCV* approach where we subtracted 1 from both the numerator and denominator; in the *EDVV* we only subtract 1 from the numerator, owing to the fact that a MAD between the same algorithm equals 0.

The resulting matrix $W=\{w_1,w_2\dots,w_n\}$ is formed with the specific weights for each algorithm.

2.3.2 Putting it all together

The last phase of our general approach uses the weights W produced by either of our proposed variants: *EDCV* or *EDVV*.

Algorithm 2.3 The *EDVV* approach for joining outlier scores

output: Return matrix of weights $W = \{w_1, w_2 \ldots, w_n\}$

procedure `EDVV()`
  Compute a matrix (D) of mean absolute deviations (MAD) between the standardized output scores F;
  For each technique, produce w_n as the average of its corresponding column of deviations D_m;
  for *each n in T* **do**

$$Ofinal = \frac{\sum_{m=1}^{T} Dmn}{T-1}$$

  end
end procedure

Algorithm 2.4 Final averaged output after applying the corresponding votes and weights

output: $Ffinal$

procedure `FINAL OUTPUT()`
  for *each i in m* **do**

$$Ffinal = \frac{\sum_{j=1}^{T} (F(i,j) * V(i,j) * W(j)}{T}$$

  end
end procedure

The final process is displayed in Algorithm 2.4. First, we calculate the product of each of the standardized scores F and their corresponding votes in matrix V, then the resulting values are updated by applying the weights W obtained by either *EDCV* or *EDVV*. Finally, the updated scores from each algorithm are simply added together and divided by the size of T.

The output of this last phase is a vector of size m (number of observations) with the weighted and voted scores of all the algorithms of the ensemble. These final scores have two main advantages over a simple averaging approach: first, they increase the relative distance between potential outlier and inliers, and second, they promote the outputs of the algorithms exhibiting the better expected performance.

In the following section, we present the experiments using real world datasets comparing the proposed approaches with 3 similar approaches: simple averaging, feature bagging *Cumulative Sum* and feature bagging *Breadth First*).

2.4 Experiments and evaluation

2.4.1 Methods and parameters

For our experiments, we compare the results of our approach with those of simple averaging, feature bagging *Cumulative Sum* and feature bagging *Breadth First*. We set the number of iterations for feature bagging to 50, while for simple averaging, *EDCV* and *EDVV* we used 4 iterations (one for each algorithm).

Feature bagging in its two variants (*Cumulative Sum* and *Breadth First*) uses only a single algorithm applied n times. The authors report their results using LOF as the single algorithm of their ensemble, thus when comparing our results with those of feature bagging, we also use LOF.

We set the number of algorithms in both approaches (*EDCV* and *EDVV*) equal to 4. The algorithms used in our ensemble are: LOF, k-means clustering, hierarchical clustering and a modified boxplot method.

We use LOF as the technique with the expected best performance in our ensemble and the rest is formed with techniques whose performances are not expected to be better or significantly better than those provided by LOF.

The choice of the algorithms composing the ensemble was made in order to obtain a diversified set; by diversified we refer not only to the type of technique (distance or density-based), but also to the quality of the results. In this way, the resulting set consists of different types of algorithms with different performances. The idea is to simulate a real world scenario where it

is not possible to know in advance which technique is the more suitable for the dataset under study.

Where possible we use the default values of each algorithm, and in the case of clustering and LOF that need some adjustment in their parameters, we do not try to tune the configuration values to the specific domain or dataset. Instead, we use the same parameters with all the datasets; obviously tuning these values would result in a better overall performance, but we are simulating a scenario where there is no additional information about a particular dataset.

The goal in our experiments is to mimic a real estimation of the performance of the ensemble methods and not the performance of perfectly tuned outlier detection algorithms. Differently from the experiments performed by the authors of feature bagging who used the labels for the inliers (normal instances), we do not suppose the existence of labels, given that our experiments are based on a completely unsupervised approach. Despite this, we acknowledge that the inclusion of labels for the inliers will increase the performance of the algorithms and thus that of the ensemble.

Our results are also compared to a simple average of the scores of each algorithm, which surprisingly gives interesting results.

To choose the configuration values for LOF and$k$-means, we follow the suggestions from (Hartigan & Wong, 1979; Breunig *et al.*, 2000). For LOF, the parameter indicating the number of neighbors was set to 20; this decision was made by averaging the author's suggestion to use a value between 10 and 30 in the absence of more knowledge about the dataset under study. For the k-means clustering algorithm, we set the number of centers to eleven (k=11). The remaining two algorithms, hierarchical clustering and modified box-plot, were used with their default values.

2.4.2 Datasets

The datasets were selected based on: (a) real world problems, (b) different proportions of classes, (c) different number of variables and (d) used by previous and similar research on outlier detection. Table 2.1 gives the characteristics of the selected datasets located on the UCI machine learning repository (Lichman, 2013).

For the breast cancer and ionosphere datasets, we did not perform any modification; we simply took the smallest class as the outlier class, and the rest as the normal (inlier) class. With the former dataset, the smallest class represents a classification of malignant cell nuclei, whereas the bigger class represents the benign case. The latter dataset consists of measures from high-frequency antennas detecting free electrons in the ionosphere; the majority class is composed of those measures representing some structure in the ionosphere, and the minority class by those cases where there is no evidence of structure formation in the ionosphere. For the satimage dataset, we use the smallest class as the outlier and merged the rest to be considered as the normal class. In this dataset, the classes represent multispectral values of pixels in a satellite image. When performing experiments on lymphography, we selected classes one and four (less than 5%) to be the outlier class and used classes two and three as the normal class.

Table 2.1 Datasets characteristics (Cl=Classes, At=Attributes, O=Outliers, I=Inliners)

Dataset	Cl	At	O	I	O (%)	Modifications
Breast cancer	2	32	212	357	37.26	Class 2 v/s. 1
Ionosphere	2	34	126	225	35.90	Class 2 v/s. 1
Lymphography	4	18	6	142	4.05	Merged class 1 & 4 v/s. rest
Satimage	7	36	626	5809	9.73	Small class v/s rest
Ann_thyroid (average)	3	21	73-177	3178	2.24-5.28	Each class v/s. 3
Shuttle (average)	6	9	2-809	11478	0.02-6.58	Classes 2,3,5,6 & 7 vs. class 1

To increase the number of available datasets, we used a procedure commonly used in similar studies (Lazarevic & Kumar, 2005; Joshi & Kumar, 2004), which consists in the adaptation of datasets not directly related with the problem of outlier detection. The procedure consists of transforming a multivariate problem into a two class problem in two steps: first, we identify the smallest class or a subset of the smallest classes, and consider them as the outlier class, then, the majority - or the rest of the classes - are merged and used as the normal class. Following this method, we formed 7 additional datasets based on ann_thyroid and shuttle datasets. Accordingly, for the ann_thyroid dataset, which contains three classes, the smallest two are related with hyperfunction and subnormal function (less than 10% of the dataset), and a third not hypothyroid class (normal condition); in this case, we produced 2 datasets by using each one of the minority classes in turns as the outlier class versus the normal condition.

Finally, for the shuttle dataset containing 6 classes, we selected class 1 (80% of the data) as the normal class and each of the remaining 5 classes in turns as the outlier class, obtaining 5 additional datasets.

Table 2.2 AUC (area under the curve) for simple averaging, feature bagging (FB) cumulative sum, feature bagging (FB) breadth first and our proposed approaches EDCV and EDVV.

Dataset	**Simple Average**	**FB cum.sum**	**FB Breadth first**	**EDCV**	**EDVV**
Breast cancer	0.8439	0.6475	0.6695	0.8489	**0.8609**
Ionosphere	0.8711	0.8654	0.8824	0.8916	**0.8980**
Lymphography	0.9871	0.9871	0.9765	**0.9894**	**0.9894**
Satimage	0.6439	0.5149	0.5079	**0.6517**	0.6326
Ann_thyroid (average)	0.7331	0.7081	**0.8360**	0.7501	0.7485
Shuttle (average)	0.9955	0.9133	0.9096	**0.9972**	0.9970

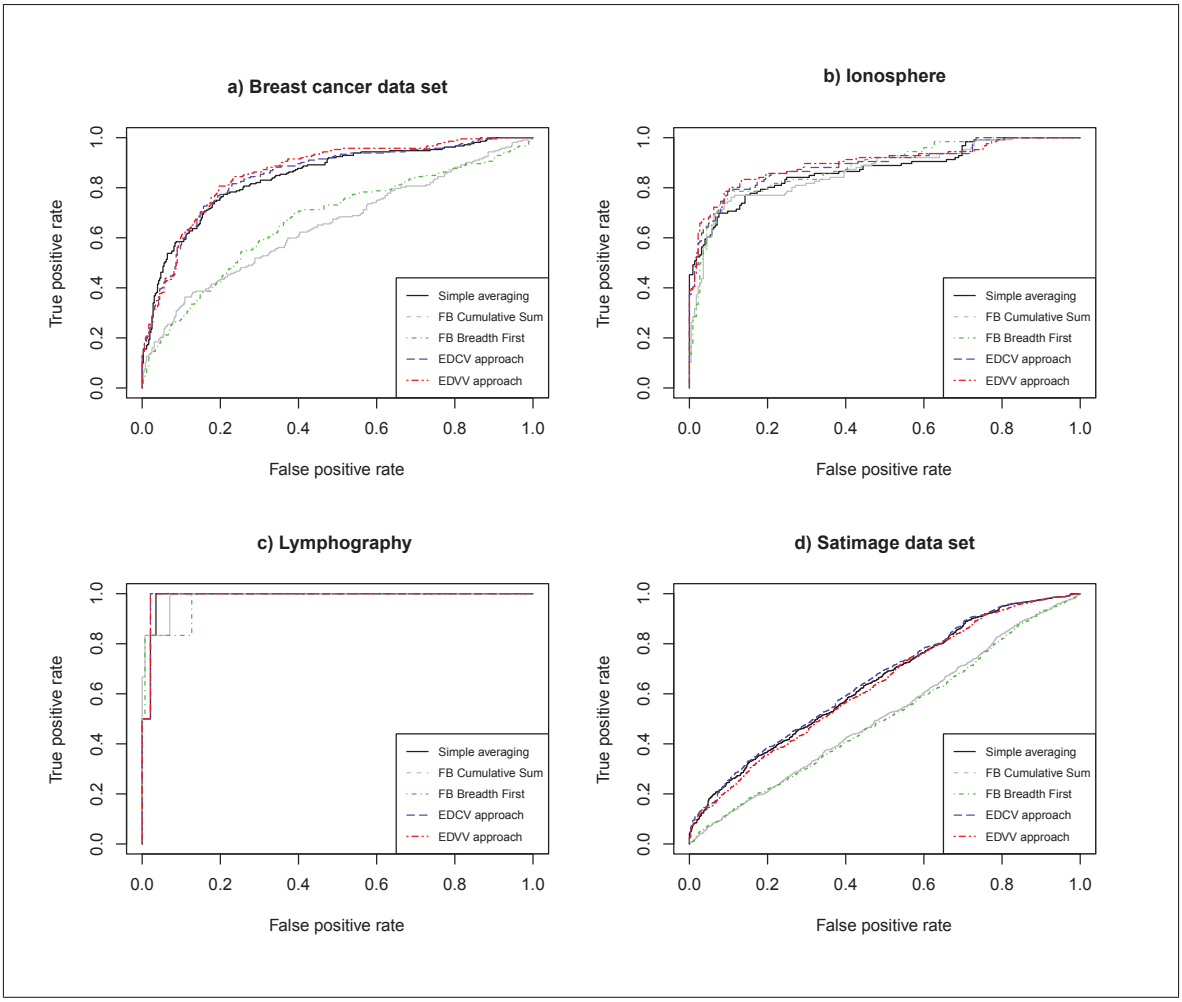


Figure 2.1 ROC curves for LOF, Feature bagging and FBSO in Segmentation, Satimage, Waveform and Gisette datasets.

2.4.3 Results

The results of our experiments on the resulting 11 datasets were presented in Table 2.2. The ROC curves for simple average, feature bagging (*Cumulative Sum* and *Breadth First*), *EDCV* and *EDVV* were displayed in Figure 2.1. In the case of the ann_thyroid and shuttle datasets that were adapted to a binary class problem, the results were presented using the average of the AUC over the artificially produced datasets; their ROC curves were not presented for space reasons. For breast cancer, ionosphere, lymphography and satimage datasets, we presented both the AUC and the computed ROC curve.

Table 2.2 showed that both *EDCV* and *EDVV* outperformed simple average, FB *Cumulative Sum*, and FB *Breadth First* in almost all the datasets, the exception being the ann_thyroid dataset, where FB *Breadth First* showed better results; the main reason for this behavior is the dependence of *Breadth First* on the order in which the outputs of the algorithms are presented. Nevertheless, the authors of the *Breadth First* approach do not contemplate a procedure to sort these outputs and consequently, this approach relies on a random order, in the case of ann_thyroid the resulting random order was favorable to *Breadth First*. Despite that, both *EDCV* and *EDVV* showed better performance than FB *Cumulative Sum* and simple averaging.

As expected the worst performance for all algorithms was with the datasets adapted to a binary class problem. This is understandable since the union of different classes produced a single class with different distributions that are very difficult to detect by the individual algorithms of the ensembles. However, even on the artificially generated datasets, *EDCV* and *EDVV* offered an improved performance compared with the rest of the approaches. The advantage of *EDCV* and *EDVV* is that they do not assume an exceptional and constant good performance of the algorithms over all the different types of datasets, but instead, assign weights to the algorithms based on their performance on each dataset in particular.

Surprisingly, a simple average of the scores produced by the outlier detection algorithms gave a constant good performance.

More constant improvements in *EDCV* and *EDVV* were found in the datasets originally designed for a binary classification (Figure 2.1). Table 2.2 showed that the AUC for both approaches (*EDCV* and *EDVV*) was better in the datasets of breast cancer, ionosphere, lymphography and Shuttle. Besides ann_thyroid, satimage was an exception where only *EDCV* had higher AUC than the rest of the ensembles.

2.5 Conclusions

In this paper, two novel and completely unsupervised ensemble approaches for combining the output scores of different outlier detection algorithms were presented: ensemble of detectors

with correlated votes (*EDCV*) and ensemble of detectors with variability votes (*EDVV*). Experiments on several popular real life datasets suggested that both approaches can achieve better performance than similar methods. Also, it is worth considering that our results were obtained using only 4 iterations of the ensemble, while for feature bagging we set the number of iterations to 50.

These improvements were related to the fact that *EDCV* and *EDVV* do not make presumptions about the performance of the algorithms until they are capable of comparing their outputs; thus the advantage is that both approaches are not expecting an exceptional and constant performance from all the algorithms on different types of datasets. Moreover, not expecting a constant performance of the algorithms allows for the inclusion of different types of outlier detection algorithms. While similar approaches like feature bagging *Cumulative Sum* and feature bagging *Breadth First* introduce diversity through variation on the search space, *EDCV* and *EDVV* attempt to ensure diversity by using different types of algorithms, which results in a more widely applicable approach.

Despite this, we consider that our results can be improved by using feature bagging variation of the search space as a way to deal with noisy attributes. In future work, we will attempt to address this possibility.

CHAPTER 3

BAGGED SUBSPACES FOR UNSUPERVISED OUTLIER DETECTION

José Pasillas[1], Sylvie Ratté[1]

[1] Département de génie logiciel et des technologies de l'information , École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Paper published in the journal « Computational Intelligence » from Wiley, July 2016.

ABSTRACT

In many domains, important events are not represented as the common scenario, but as deviations from the rule. The importance and impact associated with these particular, outnumbered, deviant, and sometimes even previously unseen events is directly related to the application domain (ex. breast cancer detection, satellite image classification, etc.). The detection of these rare events or outliers has recently been gaining popularity as evidenced by the wide variety of algorithms currently available. These algorithms are based on different assumptions about what constitutes an outlier, a characteristic pointing towards their integration in an ensemble to improve their individual detection rate. However, there are two factors that limit the use of current ensemble outlier detection approaches: first, in most cases, outliers are not detectable in full dimensionality, but instead are located in specific subspaces of data; and second, despite the expected improvement on detection rate achieved using an ensemble of detectors, the computational efficiency of the ensemble will increase linearly as the number of components increases. In this article, we propose an ensemble approach that identifies outliers based on different subsets of features and subsamples of data, providing more robust results while improving the computational efficiency of similar ensemble outlier detection approaches.

3.1 Introduction

Outlier detection algorithms are designed to find deviations that represent crucial events in a variety of applications domains. Differently from other similar data mining approaches such as ensemble clustering (Gionis *et al.*, 2007; Ghosh & Acharya, 2011), where the main task consists in finding the prevalent classes, outlier detection algorithms are designed to detect those observations that deviate from the normal behavior. Besides being characterized as deviations, many more definitions of outliers exist in the literature (Grubbs 1969, Barnett and Lewis 1994), each of them based on different assumptions about the data. In outlier detection, the cost of misclassifying a true positive observation is far greater than the cost of misclassifying a true negative observation, as is the case in tasks like breast cancer detection, intrusion detection, etc.

There are two main characteristics that make outlier detection a challenging task: Class imbalance and the unsupervised setting. The former, refers to the extremely imbalanced proportion of outliers when compared with that of inliers. Any algorithm not taking into account this imbalance will assign the same weight to both classes and can therefore achieve high accuracy by simply assigning all observations to the predominant inlier class, however doing so will lower the detection of the few but relevant outliers. The latter, indicates the absence of labeled data for both outlier and inlier classes. While it is true that some datasets offer labels at least for the inlier class (semi-supervised), which in turn can be used for training the algorithm, it is also possible that this semi-labeled data can be contaminated with the presence of some outliers disguised inside the inlier class, with the direct results of a bias in the training phase. Throughout this paper we will focus on the unsupervised scenario, with no further information about any of the two classes.

Ensembles methods are used to improve the detection rate and robustness of a single algorithm. Their use has been extensively studied in the ensemble clustering literature (Bickel & Scheffer, 2004; Muller *et al.*, 2012a). Compared with the more mature fields of classification and clustering, where there is a wide variety of ensemble approaches, in ensemble outlier detection, the

number is very limited. This lack of approaches is due, mainly, to the difficulties associated with the characteristics of the datasets with which outlier detection operates, like class imbalance and unsupervised scenario. Despite these considerations, unsupervised ensemble outlier detection is emerging as an important research field (Aggarwal, 2013a), which provides a way to improve the applicability and performance of outlier detection algorithms in the absence of ground truth. The intuition behind an ensemble of outlier detectors is that the combination of diverse and accurate algorithms or even variations of the same algorithm can complement each other and provide an improved result. Thus, an important factor to consider when building an ensemble is to use accurate components whose results are uncorrelated (diversity of results); despite the clear importance of having accurate algorithms, the combination of accurate algorithms with identical results will not improve the overall detection rate.

Nearest neighbor algorithms based on locality tend to produce outlier scores adapted to the variations in the local density around the query instance(Schubert *et al.*, 2014b). An iconic outlier algorithm based on local densities is LOF. "The central contribution of LOF and related methods is hence to enhance the comparability of outlier scores for a given dataset" (Schubert *et al.*, 2014b); however, this adaptability also makes of LOF an unstable algorithm, with high variance, as its scores will vary depending on the sample use to compute the distances to the query instance; this instability, while is not desirable in a single execution of an algorithm, in an ensemble approach is indeed desirable, as it can provide to the ensemble with a source of diversity, and hence a reduction in the global variance when combining the output of each ensemble member. Then, throughout this paper we will use LOF as a baseline with which to compare our proposed approach which in turn uses LOF as its base algorithm.

Besides accuracy and diversity, the execution time is a crucial factor when building an ensemble. This is a problem present not only in ensemble outlier detection but in any ensemble approach; the execution time of the ensemble is directly related to the number of ensemble components, for example, a 10 components ensemble will show an execution time about 10 times higher than that of the base method. In scenarios where there is a time constraint, the

number of ensemble components needed to obtain diversity and accuracy has to be carefully considered.

In this paper, we propose an unsupervised ensemble outlier detection approach that deals with these problems: diversity of inputs and global execution time. This approach induces diversity by combining the outputs of a single outlier detection algorithm (LOF), fed with random subsamples of data and random subset of features(in different iterations). This use of random subset of data provides not only an improvement in detection rate but also in execution time. Similar approaches in the literature induce diversity either by subsampling the data for estimating the density around a specific data point (Zimek *et al.*, 2013) or by using different subset of features for each iteration of the ensemble (Lazarevic & Kumar, 2005). For this work we build upon these two approaches.

Through experimental evaluation on real world datasets, we demonstrate that our proposed approach improves the detection rate and execution time when compared to other ensembles approaches for outlier detection, and that under certain conditions, can perform in an execution time similar to that of a single outlier detection algorithm.

The rest of the paper is organized as follows: section 3.2 provides a background about outlier detection and unsupervised ensemble methods for outlier detection; section 3.3 introduces our approach in detail; section 3.4 illustrates some experiments on synthetic and real life datasets, and finally, section 3.5 concludes our research and discusses the scope for future work.

3.2 Related work

The notion of what constitutes an outlier is dependent on the application domain. Outliers can be induced by different mechanisms like malicious activity, errors in the generative process, or they simply represent outlying but valid observations. Despite their nature, a common characteristic is that they represent interesting information for the user. The volatility of the notion makes the identification of outliers a very difficult task. To cope with this, a wide variety of techniques for outlier detection algorithms (Chandola *et al.*, 2009) has been proposed:

classification, clustering, and statistical methods, nearest neighbor based methods, information theoretic and spectral approaches.

A common group of techniques for the detection of outliers in an unsupervised scenario are those based on nearest neighbors (k-NN). The use of k-NN algorithms for the detection of outliers is based on the assumption that outlying observations show a relative larger distance to its nearest neighborhood when compared to that of a normal instance. A variation of this is the use of the relative density of the neighborhood in order to compute the outlier score. This group of techniques considers that outliers are located in low density regions, while inliers are in high density regions; this relative density of a data point is used as an outlier score. An seminal example of this type of approach is the Local Outlier Factor (LOF) (Breunig *et al.*, 2000) algorithm, which assigns outlier scores depending on both the reachability of a data point and the relative density of its neighborhood. This approach estimates the density of the neighborhood of each observation, assigning them an individual score. First LOF ascertains a sphere centered at a particular observation covering its k nearest neighbors. The local density will then be computed by dividing k by the volume of the sphere.

NN techniques have the advantage of providing results easy to interpret, are capable of handling differet types of features, are capable of dealing with noise in the data and the model can be updated as more data arrived.(Kelleher John D., 2015)

LOF is heavily influenced by the relative density of its neighborhood; thus, computing this density iteratively with random sets of observations can provide different and potentially complementary results, which in turn can be used to reach diversity and reduce the variance when building an ensemble approach.

3.2.1 Ensemble outlier detection

Before the concept of outlier ensembles was explicitly applied by Lazarevic (Lazarevic & Kumar, 2005), different approaches for outlier detection were already using implicitly the idea

(Aggarwal, 2013a), but it was hidden deep inside the outlier algorithm and not formally recognized as an ensemble approach.

Aggarwal (Aggarwal, 2013a) proposes a categorization of ensemble outlier algorithms based on two characteristics: component independence and constituent components. The former, refers to whether the components work in a sequential order (sequential ensembles) or whether they can function independently of one another (independent ensembles). The latter, addresses the composition of the ensemble either with the use of the same algorithm, working on different subspaces of the data (data-centered ensembles), or the use of different algorithms (model- centered ensembles). It is important to note that using different parameters for the same algorithm can also be considered as a case of modeled centered ensemble.

There are four main issues to consider when building an ensemble of unsupervised outlier detectors, Zimek exposes (Zimek *et al.*, 2014) three of them: first, how to measure accuracy in the absence of labels; second, how to measure the diversity of the models and third, how to combine these models. A fourth issue is the ability of the ensemble to search for outliers in full dimensionality, with a mixture of contributing and noisy features.

Ensemble outlier detection, like any other ensemble of classifiers, needs a combination of algorithms that are accurate while at the same time diverse. Accuracy and diversity are needed to produce an improved result over the base algorithm; however it is important to establish the right balance of both. Measuring the accuracy of each ensemble member in an unsupervised scenario is a challenging task, without labeled data it is not possible to perform a typical external evaluation of the algorithms by comparing their outputs with a ground truth labels set. Different from accuracy , there are different methods for inducing and measuring diversity, Zimek (Zimek *et al.*, 2014)proposes a classification of five groups based on: different types of subsets of features (Lazarevic & Kumar, 2005; Keller *et al.*, 2012; Muller *et al.*, 2012b), different subsets of objects (Zimek *et al.*, 2013), isolation forests (Liu *et al.*, 2012), parameter variation (Schubert *et al.*, 2014a; Jing & Pang-Ning, 2006), and different set of models (Kriegel *et al.*, 2011; Nguyen *et al.*, 2010; Schubert *et al.*, 2012).

As explained in (Aggarwal & Sathe, 2015) the use of subsampling will provide ensemble members with higher variance and bias when compared with the execution on full dimensional space. However, the variance of each detector will contribute as a diversity source when combined in an ensemble scheme. This bias-variance trade-off is of particular importance in the ensemble case, as the bias and variance induced will increase as the sample size decreases. Clearly, an algorithm using only a subset of the data will have an inferior performance when compared with its equivalent on full dimensionality. However, an ensemble approach can use the diversity found in each algorithm to build its final set of scores by averaging the variable set of outputs. Then, what is a disadvantage in a single detector, provides in an ensemble scenario a valuable source of diversity.

An ensemble outlier detection approach with the right combination function for a set of diverse and accurate results, still has to deal with the high dimensionality of most real world data. Outlier detection in high dimensionality is a complex problem, as the data becomes sparse the notion of proximity is no longer meaningful, and even normal observations can show an outlier behavior (Hinneburg *et al.*, 2000; Aggarwal *et al.*, 2001). Also, as the number of dimensions increases the complexity of searching for outliers in all possible subspaces increases. The number of possible unordered subspaces is equal to 2^d-1, where d is the number of dimensions. In low dimensional data this is not a problem as it is possible to search for outliers in all possible combinations of attributes, but as the dimensionality increases, so does the complexity time. For high-dimensional cases, it is thus infeasible to analyze each possible subspace. For example, with d=2, there are 2^2=4 subspaces to analyze, but when d=20, the number of subspaces is equal to 2^{20}=1,048,576. Studies of unsupervised outlier detection on high-dimensional data can be found in (Hinneburg *et al.*, 2000; Aggarwal *et al.*, 2001; Aggarwal & Yu, 2001).

However, Zimek (Zimek *et al.*, 2012) point out that the main concern is not only the increasing number of dimensions, but also the existence of too many irrelevant or noisy attributes that do no contribute to the identification of outliers, and that can mask the interesting observations. In the same sense, the behavior of some outliers can be detectable only in a specific subset of

dimensions (Xuan Hong *et al.*, 2014), and some outliers in the same datasets can be detectable only with different specific combinations of attributes.

The removal of noisy or useless features is straightforward when expert's knowledge is available. However, in outlier detection the existence of labeled data is scarce; the most common case is the absence of labels for both inliers and outliers. Moreover, the removal of a dimension could hinder the detection of outliers located in specific subspaces. In these cases the inclusion of an irrelevant dimension is less damaging that the exclusion of a relevant dimension. Approaches based on multiple sets of subspaces can avoid losing these valuables dimensions, while contributing to the robustness of the results. Throughout this paper we will refer to a dataset in terms of its dimensionality d (number of variables) and its size N (number of observations).

3.2.2 Feature bagging

Feature bagging for outlier detection (Lazarevic & Kumar, 2005) computes the outlier scores in a series of T rounds. In each round, it computes the outlier scores using a different set of features; the authors recommend that the number of attributes vary between $d/2$ and d-1. The output of this approach is a set of outlier scores computed using different set of attributes.

The main purpose of using this method is to induce diversity in the ensemble, not by using different types of algorithms, but by varying the dimensions used when computing the outlier scores. Despite its advantage to induce diversity, its complexity time still depends on the complexity time of the base algorithm and the number of iterations of the ensemble.

3.2.3 Subsampling

Zimek (Zimek *et al.*, 2013) proposes the use of subsamples of data to feed an outlier detection algorithm. Computing outlier scores on subsamples improves the time complexity of the ensemble. This ensemble method, coupled with an outlier detection algorithm based on relative densities like LOF, can provide not only a faster processing time, but also a diverse set of re-

sults. It is important to note that this method does not simply takes random subsamples of data to compute the score for the points in that sample; doing so would not assign scores for all the observations. Instead it uses these random subsets to compute the nearest neighbors and then the density estimates for each observation in the dataset.

3.3 Feature Bagged Subspaces for Outlier Detection (FBSO)

We propose a novel unsupervised ensemble outlier detection approach: Feature bagged subspaces for outlier detection (FBSO). The target of FBSO is to improve the detection rate of outliers while maintaining a low execution time.

To avoid increasing the prevalent sparsity of the ensemble outlier detection literature, we use Aggarwal's classification (Aggarwal, 2013a), alluded to in the previous section. Our approach can therefore be classified as an independent and data centered ensemble. The former is due to the independence of the decision of each ensemble component, meaning that each outlier algorithm is not affected by the performance or decisions of the others. The latter, is explained by the source of diversity of the ensemble, which is not induced with the use of different algorithms, but instead with the use of subsets of features and subsamples of data. An interesting result of using subsets of data, is that this way of inducing diversity can provide not only an improvement on detection rate, but also on the overall complexity time of the ensemble (Zimek *et al.*, 2013).

FBSO induces diversity in the ensemble in two ways : Feature bagging (Lazarevic & Kumar, 2005) and subsamples of data (Zimek *et al.*, 2013). While feature bagging provides the ensemble with different number and sets of features at each iteration, subsampling computes outlier scores based on different subsamples of data.

The following three subsections describe in detail the FBSO process. The selection of subspaces and subsamples of data are presented in section 3.3.1 and 3.3.2, respectively. Section 3.3.3 describes how to use the subset of dimensions and subsamples of data to produce a unified set of outlier scores.

3.3.1 Lower dimensional spaces

In our proposed approach, the search for outliers is performed in lower dimensional spaces derivated from the full dataset. The use of subspaces offers a robust set of results, avoiding two main problems when searching for outliers in full dimensionality: first, the performance degradation of density-based algorithms with increasing dimensionality and second, the outlier behavior of some observations detectable only in specific subsets of dimensions. Due to the unsupervised nature of outlier detection, the selection of the most relevant set of dimensions cannot be based on the use of ground truth labels. Also, searching for all possible subspace combination is not feasible in high dimensionality. Instead, we are using random sets of subspaces (feature bagging) to improve the chances of detecting lower dimensional outliers.

The subspaces F in FBSO are obtained by randomly selecting features (without replacements) from the original dataset D, being $F=\{(f_1),(f_2)\ldots(f_t)\}$; f_t is a set of attributes of random size between $d/2$ and d-1, where d is the dimensionality of the data.

The sets in F guides the search for outliers in lower dimensional subspaces while providing a mechanism to reduce variance by inducing diversity in the ensemble. However, despite operating in a lower dimensional space, the complexity time, the bias and the variance of the ensemble continues to be heavily influenced by the data size (number of observations). Working on a lower dimensional space inherently affects the bias-variance of the base algorithms, while the ensemble benefits with the variability found in each detector operating in lower dimensions, the bias of the algorithms will increase depending on how many of the original dimensions are relevant to differentiate between outliers and inliers (Aggarwal & Sathe, 2015).

3.3.2 Subsampling for density estimation

Complexity time of a density-based outlier detection algorithm, like LOF, is not only dependent on the dimensionality of the data, but it also largely depends on the number of observations. Then, besides the capability of FBSO of working on lower dimensional spaces, it also uses subsamples of data, which, as mentioned before induces bias, but produces a reduction in

the ensemble processing time and, by the diverse nature of the subsamples, a good source of diversity, decreasing the global variance of the ensemble.

The size *(s)* of the subsamples can be set between 0.1 and .9; a sample of .1 corresponds to a subsample whose size is 10% of the original data, and correspondingly, a data subsample of .9 corresponds to a sample of 90%. We use these subsamples to obtain the density estimates of each observation in D. Density estimates are computed with LOF in different iterations, using different sets of neighborhoods for each observation.

3.3.3 Feature bagged subspaces

Our proposed approach uses two different mechanisms to induce diversity, hence reducing the global variance: random samples of data to compute density estimates, and random variation of the available dimensions.

The general algorithm is depicted in Algorithm 3.1, where D represents the whole dataset, d, the number dimensions, and s and T are user-specified parameters corresponding to the size of the subsets of data, and the number of ensemble members, respectively. The parameter T determines also the number of subsamples of data.

For each ensemble iteration, first, a set of features Ft, of a random size between $d/2$ and d-1, is selected. This set of features is used to produce a lower dimensional representation, D_t of the dataset. D_t is then subsampled without replacements, to produce a subsample SD_t of size s. The resulting data representation SD_t has not only a lower dimensionality but also is a subsample of the observations of the original dataset D.

FBSO feeds LOF with different data representations SD_t for each iteration of the ensemble to produce scores O_t. SD_t is not the only data provided to LOF; if this was the case, in each iteration, only a portion of the observations will have an outlier score. Instead, FBSO uses SD_t to compute the density estimates for each observation in D_t; the density estimates are then based on a different set of neighbors, producing a more robust result. LOF is heavily influenced

Algorithm 3.1 Feature bagged subspaces

input : enmsemble members T,sample size s
output: Ensemble outlier scores $Ofinal$

procedure `FBSO(t)()`
  for *each member* $t \in T$ **do**
    Select randomly, without replacements, a set of features $F(t)$ from D of random size between $d/2$ and d-1;
    Create subset $D(t)$ from D with features $F(t)$;
    Create subsample $SD(t)$ of size s by randomly sampling (without replacements) observations from $D(t)$;
    Compute LOF scores $O(t)$ for each observation in $D(t)$ using the subset $SD(t)$ for density estimation;
  end

$$Ofinal = \frac{\sum_{i=1}^{T} O(t)}{T}$$

end procedure

by the relative density of its neighborhood; computing this density iteratively with random sets of observations can provide diverse and potentially complementary results.

Finally the sets of outliers scores O, one set O_t for each iteration of the ensemble, are joined to produce the final set of scores *Ofinal*, a single set with a unique score for each observation; each unique score consists in the average of the rows in O, the scores corresponding to an observation. This set *Ofinal* is then a combined result of the different iterations of LOF on different subspaces and subsamples of the original dataset D.

Being based on two different mechanisms to induce diversity, FBSO offers an improvement on detection rate, while maintaining an execution time lower than similar ensemble approaches. In some cases, even the execution time of FBSO is similar to a single execution of LOF in full dimensionality.

Table 3.1 Datasets characteristics

Datasets	Classes	Attributes	Noisy Attributes	Inliers	Outliers	Percentage of outliers	Adjustments
Synthetic_batch1	2	40	0	100-12000	2-240	2%	—
Synthetic_batch2	2	40	5	5000	100	2%	—
Synthetic_batch3	2	40	1-20	5000	100	2%	—
Breast cancer	2	32	—	569	21	3.56%	Class 2 v/s. 1
Lymphography	4	18	—	148	6	3.90%	Merged class 1 & 4 v/s. rest
Satimage	7	36	—	6435	62	.95%	Class 2,4 || 5 v/s rest
Waveform	3	21	—	3343	165	4.70%	Each class v/s. the rest
Segment	7	19	—	1320	99	6.97%	Class Grass, path || sky v/s.rest
KDDCup 99	2	41	—	60593	228	0.37%	U2R v/s. normal
Coil 2000	2	85	—	5474	34	.62%	Class 2 v/s. 1
Letter recognition	26	618	—	5998	240	3.85%	Each class v/s. the rest
Gisette	2	5000	—	3000	300	9.09%	Each class v/s. the rest

3.4 Evaluation

We experimented in four data scenarios, the first three are batches of 3 synthetic datasets and the last one is composed of nine real world datasets (Table 3.1).

These data scenarios are used to assess: (i) the performance with increasing data size, (ii) the detection rate and execution time with an increasing number of ensemble members, (iii) the detection rate with an increasing proportion of noisy attributes, and (iv) the performance in real world data.

3.4.1 Methods and parameters

FBSO uses the same outlier algorithm for all the iterations of the ensemble. For our experiments we decided to use LOF, this due to its tendency to show better performance than similar algorithms (Lazarevic *et al.*, 2003) and it has been used previously in the ensemble outlier detection literature with similar purposes (Lazarevic & Kumar, 2005).

We used LOF as a baseline against which to compare the results of FBSO. The scores of LOF were calculated using the complete set of features and instances in the dataset. We also compared the results of FBSO against feature bagging, an iconic approach in unsupervised outlier detection. Being LOF the base algorithm for both feature bagging and FBSO, we established the same number of k neighbors when used as a single algorithm and in both ensemble approaches.

The results obtained with LOF can vary drastically depending upon the selection of k; this single parameter required in LOF is generally chosen heuristically. For this research we chose a number of k that gives better results than random guessing to ensure that both ensemble approaches are fed with an algorithm whose output is at least superior to random selection. Probably this is the main source of bias in the ensemble, however the selection of k is application dependent.

Another parameter to take into account for both ensemble approaches is the number of ensemble members T to be used. This number is chosen as a trade-off between processing time and detection rate. A larger T tends to improve the detection rate but to degrade the processing time; and a lower value of T degrades the detection rate but improves the processing time. For FBSO the size s of the subsamples of data was set to 10

A final critical factor is the measure used to compare the results of outlier detection algorithms. While it could be possible to simply use the accuracy of each approach, its use is not recommended for outlier analysis. This is due to the highly skewed distribution of the classes in the outlier scenario, being the proportion of outliers extremely inferior (commonly less than 5%) to that of the normal instances. Consequently a simplistic approach could just classify all instances as inliers, thus obtaining a misleading high accuracy. This problem is not only related to the imbalance of classes, but also to the fact that outliers are not ordinary observations to be classified, but indeed observations that, for this task, have the highest interest for the user.

ROC curves (Receiver Operating Characteristics) overcome the problems associated with the use of accuracy as an evaluation measure for outlier detection. Thus they have been used commonly in the ensemble outlier detection literature (Lazarevic & Kumar, 2005; Zimek *et al.*, 2013). In ROC curves, the scores of outlier detection algorithms are evaluated by measuring the trade-off of the true positive detection rate versus the false positive detection rate. This trade-off is commonly represented in the form of a ROC curve (Fawcett, 2004). A commonly measure used with ROC curves is the AUC (area under the curve), used as a way to interpret with a numerical value the trade-off showed in ROC curves. The AUC measures the probability that a randomly selected positive instance will be ranked higher, by a classification algorithm, than a randomly negative one. The higher the AUC, the better the performance of the algorithm. An AUC of 1 represents a perfect classification, while an AUC near 0.5 represents a performance similar to a random classification.

3.4.2 Datasets

To test the performance of FBSO against LOF and feature bagging, we used both synthetic and real world datasets. We used the synthetic data to evaluate the performance and detection rate of the three methods with different data sizes, proportions of noisy attributes and number of ensemble members. A problem when evaluating an algorithm solely with the use of synthetic data is that this evaluation is performed with a prespecified structure without providing richer scenarios, like those found in real world data. Then, we use datasets from the UCI machine

learning repository (Bache & Lichman, 2013) to show the behavior of the methods on real world data.

For the synthetic data, we generated 3 different skewed data scenarios with 98% of inliers and 2% of outliers. Similarly to (Lazarevic & Kumar, 2005), inliers were generated from a Gaussian distribution, while outliers were generated as points far from this distribution. We established the dimensionality of the datasets in the synthetic scenarios to 40 attributes. In two of the synthetic scenarios noisy attributes were generated by iteratively reducing and incrementing the number of contributing and non contributing attributes respectively.

The specific setup for each scenario was as follows:

- ***Scenario1***. For the first test, synthetic_batch1, the size of the datasets varied between 100 and 12,000 instances to test the performance of the methods with increasing data size.
- ***Scenario2***. The second test, synthetic_batch2, was set to 5000 instances, 35 contributing and 5 noisy features. We used this more static scenario to test the performance of the methods when increasing the number of ensemble members.
- ***Scenario3***. For the third test, synthetic_batch3, the number of instances was set to 5000 and the number of noisy attributes varied between 1 and 20. With this, we tested the robustness of the methods against noise.
- ***Scenario4***. To provide a richer exploration of the behavior of the three methods, we performed experiments using real world datasets. However, a problem when selecting real world data for outlier detection is the lack of datasets specifically designed for this task. To evaluate and compare our approach we used datasets from the UCI machine learning repository (Bache & Lichman, 2013) and adapted them, as done previously in the literature (Emmott *et al.*, 2013; Zimek *et al.*, 2013), to the outlier detection problem (Table 3.1). This adaptation procedure consisted of labeling the minority class as the outlier class, and then merge the rest of the classes and label them as the inlier class. In some cases, as done previously (Keller *et al.*, 2012; Zimek *et al.*, 2013), we additionally down sampled the minority

class to diminish the proportion of outliers in the data. In the next paragraph we explain the specific modifications performed to each dataset.

For the *Breast cancer* and *Coil 2000* datasets, we labeled the minority class as the outlier class and the remaining classes as the inliers. For the *Lymphography* dataset, we merged classes 1 and 4 as the outlier class and used classes 2 and 4 as inliers. In *Satimage* and *Segment*, we selected 3 of the minority classes in turns as outliers and used the rest as inliers. For the *Waveform*, *Letter recognition* and *Gisette* datasets, we used each class in turns as the outlier class and merged the rest as the inlier class. For all datasets, except for *Lymphography*, *Kddcup 99* and *Letter recognition*, we took a sample of 10% of the outlier class. For *Lymphography*, *Kddcup 99* and *Letter recognition* we did not perform down sampling due to the already relatively low proportion of outliers. For *Kddcup 99* we selected the classes corresponding to an intrusion type U2R as the minority class. With this process of adapting datasets to the binary and highly skewed problem of outlier detection, we generated 41 datasets.

3.4.3 Results

On evaluation of the performance and detection rate of LOF, feature bagging and FBSO on artificial and real world datasets, we first explored their execution time with an increasing number of observations. Next, we analyzed the AUC and execution time as the number of ensemble members increased. Then, we explored the effect on detection rate in the presence of noisy attributes and finally, we analyzed and compared the detection rate of FBSO, LOF and feature bagging (cumulative simple average) on real world datasets.

3.4.3.1 Synthetic data

The effect of the number of instances in the AUC of LOF, feature bagging and FBSO was examined by varying the number of instances in synthetic_batch1. For Feature bagging and FBSO the number of ensemble algorithms was set to 10, for LOF a single run in full dimen-

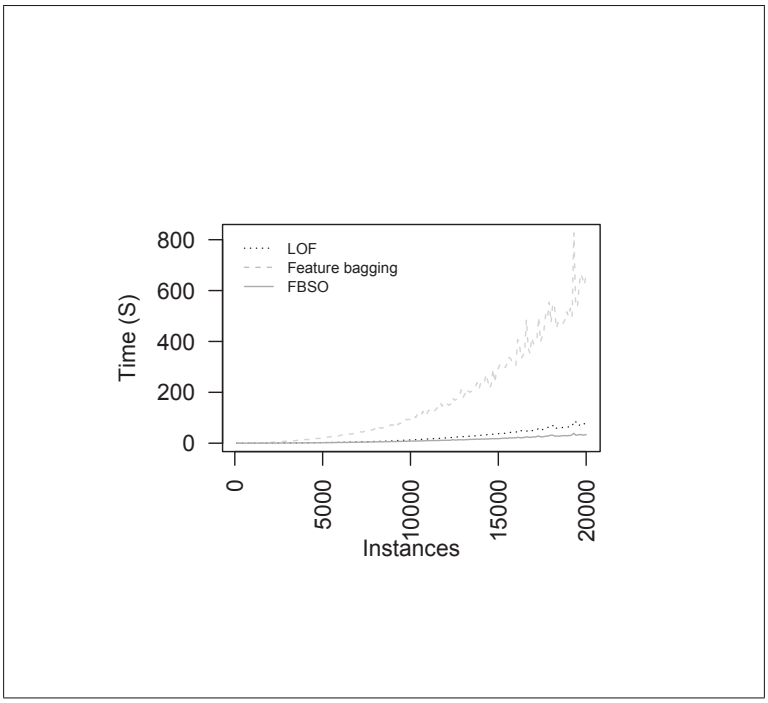


Figure 3.1 Execution time for LOF, feature bagging and FBSO with an increasing number of observations in synthethic_batch1.

sionality. It can be seen that increasing the number of instances in synthetic_batch1 had a lower impact on the execution time on FBSO than on Feature Bagging (Figure 3.1). As the number of instances increased FBSO was capable to operate in an even similar execution time than a single run of LOF on full dimensional space.

On evaluation of the effects that the number of ensemble members had in FBSO , we first evaluated the change in AUC, then we evaluated the effect on execution time. In both cases the number of algorithms varied in the range from 1 to 120 and the number of noisy attributes was set to 5 in all the datasets generated in synthetic_batch2. As the number of algorithms increased from 1 to 10, we observed a substantial improvement in AUC (Figure 3.2 (a)), however as the number of algorithms increased beyond 10 the improvement in AUC is less drastic and even showed instability. The execution time showed a more stable behavior than the AUC, increasing linearly with the number of ensemble members (Figure 3.5 (b)).

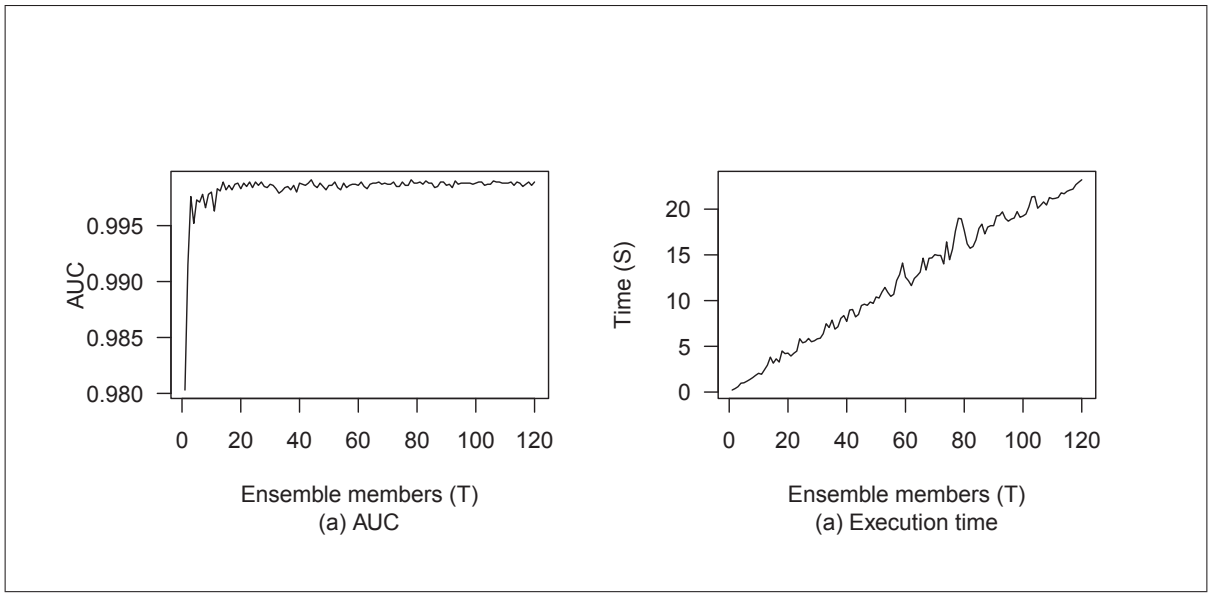


Figure 3.2 AUC and Execution time for FBSO with an increasing number of ensemble members in synthetic_batch2.

Table 3.2 AUC for LOF, Feature bagging and FBSO on real world datasets

Dataset	*LOF*	*FB*	*FBSO*
Breast cancer	0.6164	0.6579	**0.9761**
Lymphography	0.8615	0.9199	**0.98**
Satimage	0.6416	0.7074	**0.7708**
Waveform	0.6278	0.6896	**0.7150**
Segment	0.7882	0.8365	**0.9152**
KDD Cup 99	0.6221	0.6969	**0.7207**
Coil 2000	0.5475	0.5873	**0.607**
Letter recognition	0.5516	0.5558	**0.6767**
Gisette	0.6165	0.6178	**0.6708**

Measuring the effect of a variable number of noisy dimensions (synthetic_batch3), it was apparent that the AUC for the three algorithms decreased as the number of noisy dimensions increased (Figure 3.3). This deterioration was expected since the growing number of noisy attributes increased the difficulty for LOF to differentiate between outliers and inliers. However, feature bagging and FBSO, despite both being based in LOF, had a lower deterioration rate. This behavior was explained by the ability of feature bagging to deal with noisy attributes

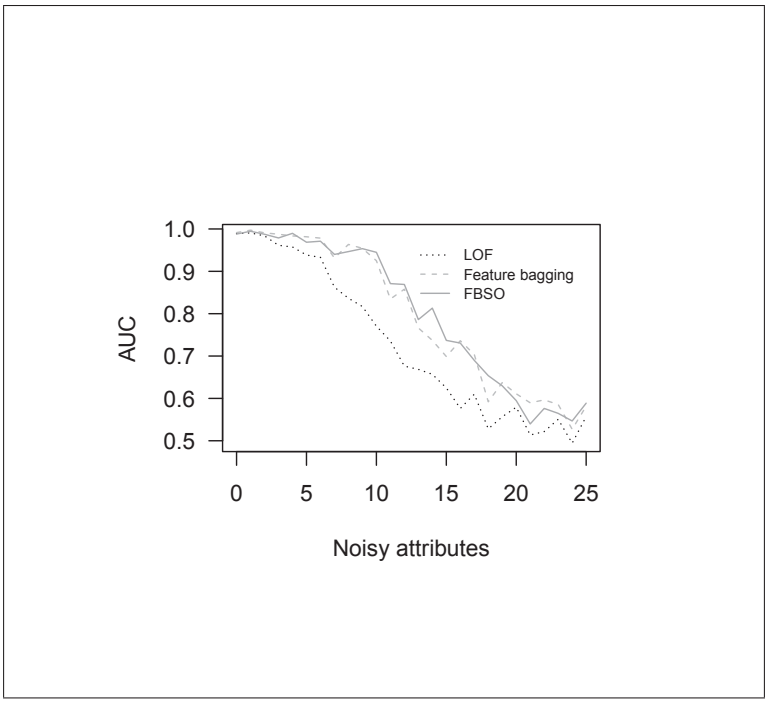


Figure 3.3 AUC for LOF, feature bagging and FBSO with an increasing number of noisy attributes in synthethic_batch3.

by using random subspaces. FBSO, being based in part on feature bagging, showed a similar behavior.

3.4.3.2 Real world data

We tested the performance of the three algorithms on datasets from the UCI machine learning repository (Table 3.1). The number of ensemble members T for feature bagging and FBSO was set to 10 for all the datasets. For *Breast cancer*, *Lymphography*, *Kddcup99* and *Coil2000* datasets, we displayed in Figure 3.4 the trade-off of true positive rate and false positive rate with the help of ROC curves. For reasons of space, we chose to present the results from the 11 datasets obtained from *Satimage*, *Waveform*, *Segmentation* and *Gisette* datasets in the form of bar charts (Figure 3.5) displaying the AUC obtained with each algorithm. The AUC of the three algorithms in all real world datasets are displayed in Table 3.2; for *Satimage*, *Waveform*, *Segmentation*, *Letter recognition* and *Gisette*, we only displayed the AUC averaged over the

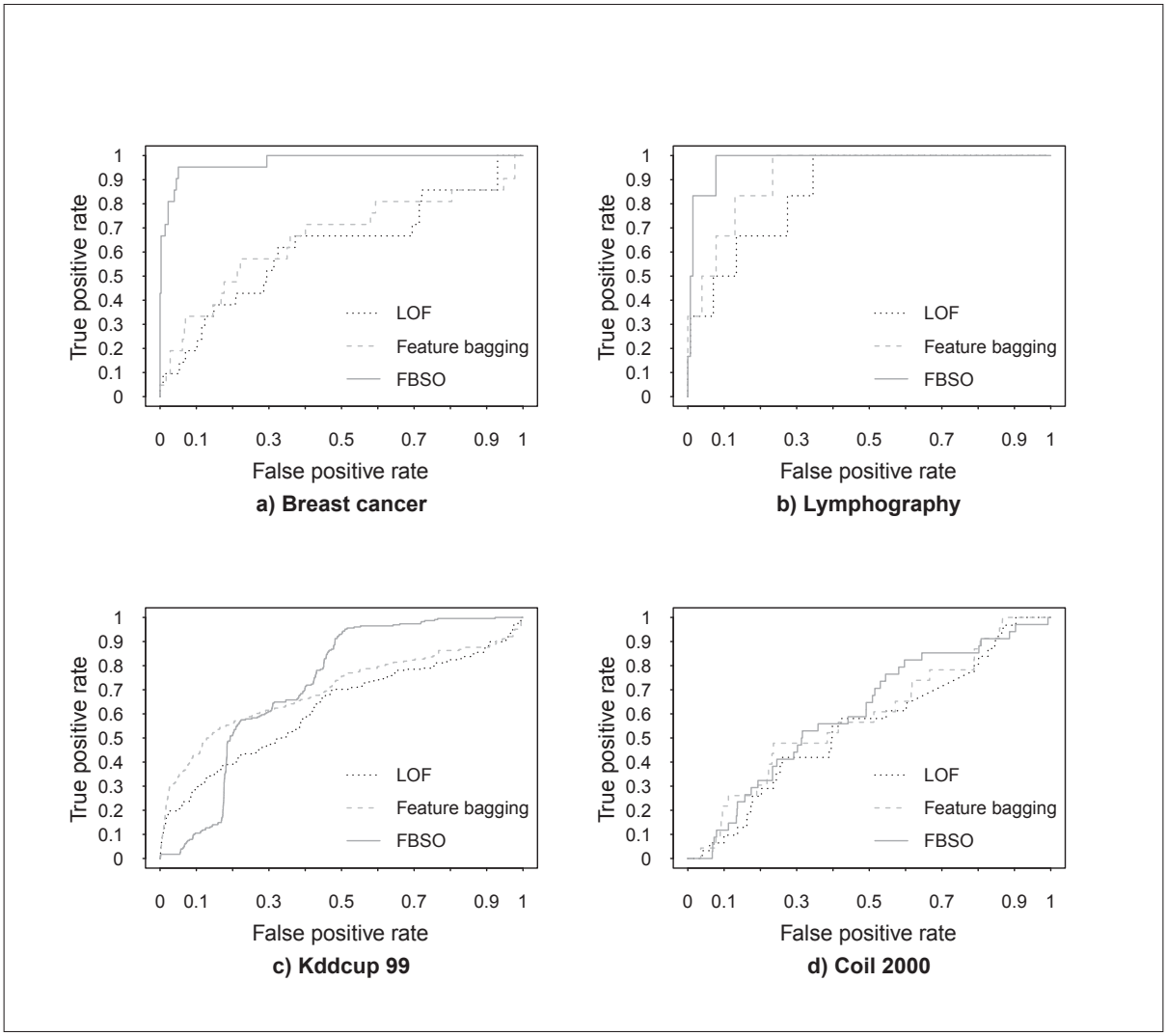


Figure 3.4 ROC curves for LOF, feature bagging and FBSO in *breast cancer*, *lymphography*, *kddcup 99* and *coil 2000* datasets.

different variations of the datasets. The AUC for FBSO and feature bagging was, in all cases, higher than that of LOF. Moreover, we can observe that the biggest increments were detected in FBSO.

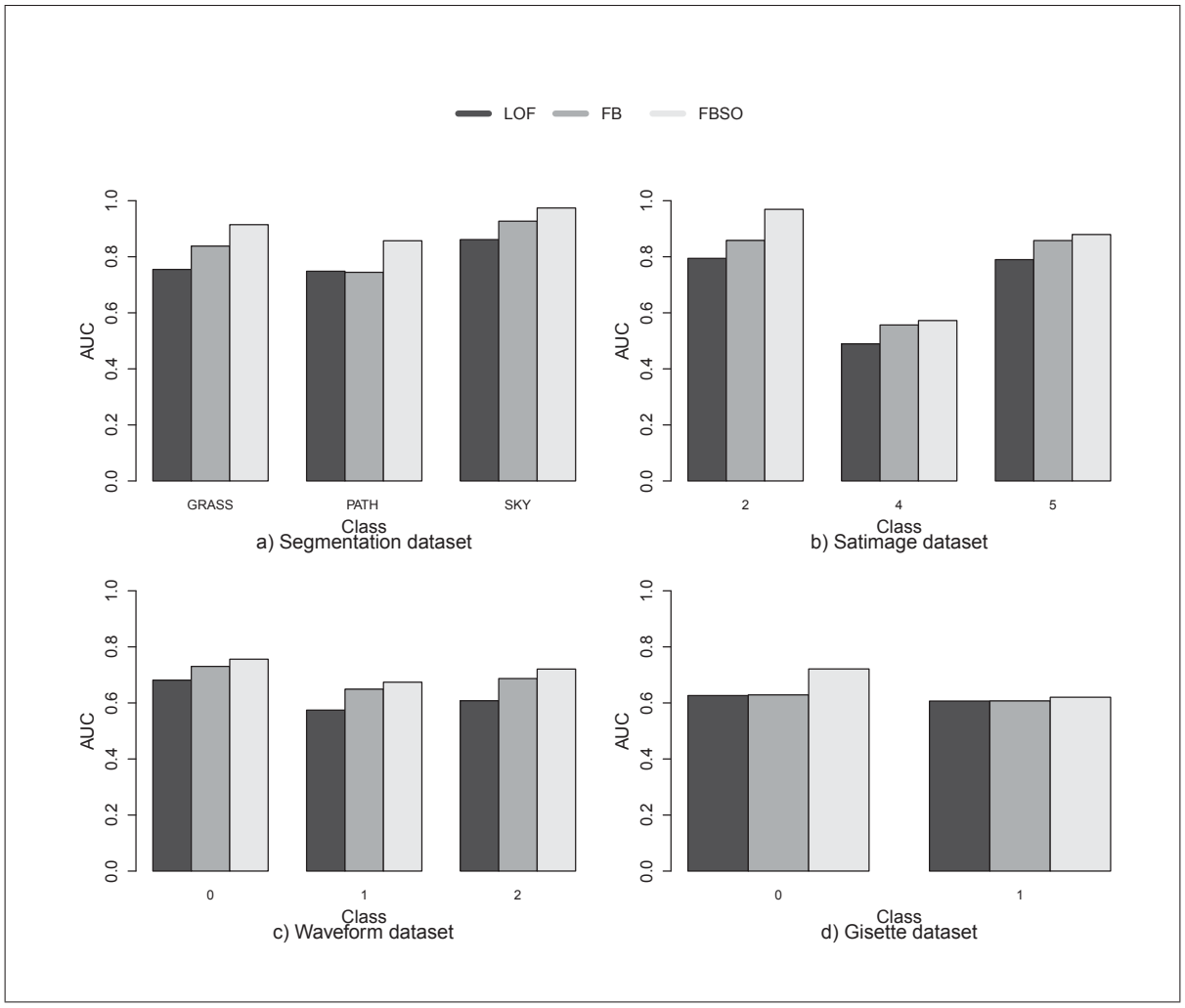


Figure 3.5 AUC for LOF, Feature bagging and FBSO in Segmentation, Satimage, Waveform and Gisette datasets.

3.4.4 Discussion

In this section we have illustrated the improvements on detection rate and execution time of FBSO when compared to LOF and feature bagging in three synthetic scenarios and real world datasets.

The results suggest that FBSO tends to have a lower execution time and higher values of AUC when compared to feature bagging. The relatively low execution time of FBSO in synthetic_-batch1 (Figure 3.1) it is not only lower than that of feature bagging, but also similar to that of

LOF on full dimensionality, this suggest that FBSO is an ensemble approach to be considered in outlier detection scenarios where the execution time is an important constraint.

The bias among the ensemble member remains constant (higher when compared with full dimensionality), but the variance tends to decrease as we add members to the ensemble. However, outlier detection deals with scenarios where an important constraint is the time. Hence, In an unsupervised ensemble outlier detection scenario, the trade-off bias-variance could be extended to a trade-off of bias-variance-execution time, where variance and execution time tend to have some degree of negative correlation, as execution time increases variance tend to decrease. Different from variance, bias will vary depending on the parameters used (application dependent), at least for the type of ensemble member we considered in this approach (LOF).

As expected, increasing the number of algorithms in synthetic_batch2 improved the AUC of feature bagging and FBSO. However, we observed in Figure 3.2 that incrementing the number of algorithms will also increase the execution time of the ensemble. This is an important trade-off to be considered by the final user. Despite the fact that finding the sweet spot is application dependent, we suggest to choose a value of ensemble members of around 10 in a scenario where the global execution time is an important concern, this ensures an execution time similar to a single execution of LOF on full dimensionality. Increasing further the size of the ensemble reported lower and variable gains on detection rate.

The detection rate of outlier detection algorithms tends to deteriorate in the presence of noisy attributes. LOF, feature bagging and FBSO are not the exception to this behavior. However, in synthetic_batch3, feature bagging and FBSO, despite both being based on LOF, have a lower deterioration rate than LOF (Figure 3.3). This behavior is explained by the ability of feature bagging to deal with noisy attributes by using random subspaces. FBSO, being based in part on feature bagging, shows a similar behavior. The AUC of the three algorithms deteriorates around 0.5 when the percentage of noisy attributes is above 50%. It is important to note that besides its relative tolerance to noisy attributes, FBSO offers the lowest execution time.

In the real world datasets, the lowest AUC values for feature bagging and FBSO was in the *Coil2000* dataset. This was expected, as the poor performance of LOF on this dataset does not contribute with quality ensemble members. Despite this, we can observe in Table 3.2 the ability of feature bagging and FBSO to improve the detection rate of LOF; this behavior is more evident in the case of FBSO. The largest increment in AUC for FBSO when compared with LOF was in the *breast cancer* dataset.

Our results showed that FBSO can be used in datasets with different dimensionality levels. However, as the dimensionality increased the performance of LOF, the base algorithm of FBSO, tends to deteriorate; which in turn, affects the performance of FBSO. This behavior is explained by the struggle of Neareast neighbor methods to differentiate between outliers and inliers as the distance between points, in high-dimensional scenarios, is increasingly indistinct. Then, although FBSO improved the detection rate of LOF in the datasets Gissette and Letter recognition, very high-dimensional datasets are not the best scenario for FBSO.

LOF's nearest neighbor calculation complexity time is $O(n^2)$, increasing with the number of instances in the dataset. The expected performance of a simple ensemble approach operating on full dimensional space is $O(n^2*T)$, where T represents the number of ensemble algorithms. Feature bagging offers a reduction on execution time by using a fraction of the available features for each iteration of the ensemble. However, this reduction is minimal and unstable. The main cost in time in feature bagging is heavily influenced by the number of observations in the dataset, showing extreme variability depending on the size of the random sets of features F_n in F, where $F=(f_1,f_2...f_n)$. The execution time of FBSO is also dependent on the random sets of features F, but the instability is decreased as the main reduction in time is achieved by the random sampling of observations. Its execution time is $O(n^2*s*T)$, where s is the sample size used for each iteration of the ensemble.

The efficiency of FBSO is due to its combined use of random samples of data and subsets of dimensions. While LOF needs to use all instances to compute the k nearest neighbors, FBSO only uses random subsamples of data, allowing a better execution time. Despite that both FBSO

and feature bagging use random subsets of features in their processes, only Feature bagging shows an unstable execution time, in all cases worst than LOF and FBSO. This instability is due to the variable and unpredictable number of features available for each iteration of the ensemble. We hypothesize that the stability in FBSO is due to its dependence not only on the subset of features but also on the random samples of data.

The results illustrate the improvement of FBSO in execution time when compared with feature bagging, and even showed a similar execution time to LOF on full dimensionality. The improvement on execution time of FBSO is accompanied with a robustness to noisy attributes. The potential of FBSO can be observed in its consistently higher AUC in real world datasets.

3.5 Conclusions and future works

In this paper we developed a new ensemble approach for unsupervised outlier detection. We analyzed that building an ensemble based on subsamples of data and subsets of features provided robustness to noisy attributes and improved the detection rate of a single outlier detection algorithm and even that of similar ensemble approaches. Moreover, using only samples of data to estimate the outlier scores had the advantage of providing FBSO with a processing time inferior to that of Feature bagging, and in some cases, to that of a single outlier detection algorithm (LOF) on full dimensionality. FBSO improved the detection rate of LOF even in relatively high-dimensional cases; however, being based on LOF, it also suffers the effects of the curse of dimensionality and its performance deteriorates as the number of attributes increased.

A consideration for further research is the possibility of using FBSO not only with LOF, but on top of different outlier detection algorithms, which we expect can improve the understanding relative to the behavior of unsupervised outlier detection methods on high-dimensional scenarios. Another open subject is the possibility of using FBSO to extract the intentional knowledge of the outlier scores (Knorr & Ng, 1999). This is an interpretation of why a particular observation is outlying. Using the information about which features contributed more for producing high outlier scores, FBSO could provide a hint about this intentional knowledge.

CHAPTER 4

ON THE BEHAVIOR OF DISTANCE MEASURES ON UNSUPERVISED ENSEMBLE OUTLIER DETECTION

José Pasillas[1], Sylvie Ratté[1]

[1] Département de génie logiciel et des technologies de l'information , École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Paper submitted to the journal « Expert Systems with Applications » from Elsevier, July 2017.

ABSTRACT

Outliers can be characterized as those observations, or group of observations, having the most discordant behavior in their data. These observations are invariably outnumbered and can either hinder a model, if they represent errors or noise in the data, or be relevant observations whose detection is critical. However, despite their infrequent and sporadic nature their potential to have a deep impact is far from trivial. The characteristics and impact of outliers is completely application dependent; accordingly, a relatively broad and diverse set of approaches for outlier detection have been proposed in the literature; as well, their behavior under different combination functions, normalization methods, types of algorithms, and data subsets or dimensionalities, has been, sporadically studied. However, the influence that different distance metrics have on the detection rate and complexity of a single algorithm or an ensemble of algorithms has thus far not been addressed in the literature; understanding how the choice of a a specific metric can perturb the behavior of distance based outlier detectors could provide some hints about variations in the detection rate and processing time when isolating factors such as data dimensionality or parameter settings. Such an insight would ease the selection of a specific distance measure depending on the inherent characteristics of the dataset (e.g., type of data, algorithm, ensemble size, etc.).

In this article we evaluated the impact on detection rate and processing time of a detector and an ensemble of outlier detectors using distinct distance metrics, increasing data dimensionality,

variations in data size, diverse parameter settings, etc; thus, unveiling the interaction outlier detector - distance metric - data. Moreover, our study provides further insights to improve current and future approaches for outlier detection.

4.1 Introduction

In 1969 Grubbs (Grubbs, 1969) established one of the most influential definitions of outliers in the literature: "An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs". This definition was subsequently extended to include not only a single point deviating from the rest of the data, but also a subset of observations appearing to be inconsistent with the remainder of that dataset (Barnett & Lewis, 1994). While there are many other definitions of outliers in the literature, they nevertheless all share the common aim of finding those outnumbered, deviant, crucial, and in some domains even previously unseen events. Despite the infrequency of outliers, their potential to have a deep impact on different application domains is far from trivial. Outlier detection application domains vary widely, and include areas such as breast cancer detection, fraud detection, satellite image identification, network intrusion detection, etc. The notion of which observations are interesting is fully dependent on the application domain; accordingly different types of outlier detection algorithms have been designed to search for outliers on distinct types of data, with each algorithm limited and oriented toward a specific assumption about what constitutes an outlier.

Two interesting outlier detection surveys covering a wide range of methodologies, applications domains and assumptions can be found in Chandola *et al.* (2009); Zimek *et al.* (2012). Moreover, other studies in the outlier detection literature cover the effects of bias and variance (Aggarwal & Sathe, 2015) , combination measures (Schubert *et al.*, 2012), normalization functions (Kriegel *et al.*, 2011), parameter settings(Campos *et al.*, 2015), attributes and/or subsamples variations (Zimek *et al.*, 2013; Pasillas-Díaz & Ratté, 2016a; Lazarevic & Kumar, 2005), combination of different types of algorithms (Nguyen *et al.*, 2010) and evaluation measures (Campos *et al.*, 2015); however, to the best of our knowledge no comprehensive evaluation has

been carried out to assess the impact of different distance measures on the processing time and detection rate of an outlier detection algorithm or an ensemble of detectors, with this work we attempt to fill this gap.

Outlier detection shares some characteristics with clustering, including an absence of ground truth for some of the classes or the use of distances measures to determine the similarity between observations; however, these similarities disappear as soon as the inherent challenges related to the identification of outliers are considered. These challenges include the highly unbalanced proportion between outliers and inliers, the absence of ground truth labels for both classes, as well as the propensity of outliers to hide in lower dimensional representations of the data (Zimek *et al.*, 2014; Aggarwal, 2013b). This singularity of the field requires a set of techniques specifically designed or adapted to outlier detection.

One of the most fundamental avenues of research in outlier detection is based on algorithms that are derived from similarity-based learning. Much like any similarity-based approach, these types of outlier dectors are essentially premised on two fundamental assumptions: first, a distance function capable of measuring the similarity among observations, and second, a feature space representation of the observations where a distance measure makes sense. An iconic outlier detection algorithm which bases its computation on a specific distance measure is Local Outlier Factor (LOF) (Breunig *et al.*, 2000), LOF takes into account not only the distance between observations, but also a local and relative density.

A comprehensive study of the impact that different distance measures have on distinct outlier detection algorithms and on ensembles of outlier detectors could provide some hints about variations on the detection rate and processing time when isolating factors such as the size of the data, the number of attributes, the parameter settings, etc. Such insight would facilitate the selection of a specific distance measure depending on the data scenario under study.

The rest of the paper is organized as follows: we discuss different distance measures commonly used in the outlier detection literature (Section 4.2). We examine the characteristics, assumptions, advantages and disadvantages of outlier detection algorithms (Sections 4.3 and

4.4). We provide a review of different evaluation measures and the reasoning behind their selection under an outlier detection scheme (Section 4.5). We provide a set of experiments on synthetic and real-world datasets evaluating the performance and detection rate of an outlier detector based on distinct distance measures (Section 4.6). We discuss how different distance measures impact detection rate and processing time, depending on the characteristics of the data or parametrization of the algorithm (Section 4.6.4). We conclude the paper and discuss the scope for future work (Section 4.7).

4.2 Distance measures

Recent advancements in the outlier detection literature are essentially oriented toward the high-dimensional scenario or to the development of new ensembles approaches. These two main avenues of research often appear merged, as new ensemble algorithms are utterly oriented toward high-dimensional data. Interestingly, most recent approaches proposed in the literature are based on some notion of similarly learning (Zimek *et al.*, 2013; Lazarevic & Kumar, 2005; Irani *et al.*, 2016).

Similarity-based approaches (Cunningham & Delany, 2007; Lin & Chen, 2010; Cha, 2007) compute the similitude between observations using a distance metric [1]. Accordingly, different distance metrics have been used in the outlier detection literature, with a focus on basic measures such as Euclidean and Manhattan distances (Birant & Kut, 2007; Knox & Ng, 1998; Angiulli & Pizzuti, 2005). Despite the almost prevalent interest in using only a couple of distance metrics, evaluating the impact that additional metrics have on an outlier detector or an ensemble of detectors could provide further hints to improve current and future approaches for outiler detection. Moreover, such study would finally unveil the interaction outlier detector - distance metric - data.

[1] Metrics are defined by 4 constrains: non-negativity, identity, symmetry and triangular inequality; however, most similarity-based approaches are also capable of using indexes, which are very similar to a metric function, but often fail to comply with one or more of the 4 metrics requirements. Throughout this manuscript, we use the term metrics to describe either metrics or indexes

The most common set of distance metrics used in outlier detection are a derivation of Minkowski distance (Eq. 4.1):

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \tag{4.1}$$

Accordingly, p=2 and p=1 correspond to Euclidean (Eq. 4.2) and Manhattan (Eq. 4.3) distances, respectively.

$$\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{4.2}$$

$$\sum_{i=1}^{n} |x_i - y_i| \tag{4.3}$$

Thus, different values of p correspond to the following metrics:

- p = 1. Manhattan distance: equivalent to absolute differences (SAD)
- p = 2. Euclidean distance: the shortest path between two points
- $p \to \infty$. Chebyshev distance: also known as chess board distance

Another commonly used measure, a weighted version derived from the basic Minkowski distances, is the Canberra distance (Eq. 4.4).

$$\sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|} \tag{4.4}$$

The vast majority of the outlier detection algorithms proposed in the literature use the Euclidean distance as the default metric; however most of the approaches can equally use any of

the aforementioned metrics. The use of different distance measures has been considered as a potential source of diversity in the ensemble process (Zimek *et al.*, 2014), offering potentially complementary views of the data. Such diversity is owed to the distinct mechanism with which each metric measures the similarity or dissimilarity between observations.

4.3 Outlier detection algorithms

Differently from classification algorithms, where, in general, the main aim is to correctly identify as many of the observations as possible members of a particular class, in outlier detection, the focus is on identifying those observations whose behavior deviates from the normal pattern. This singularity of outliers makes their presence extremely rare, and as a result, their search is invariably performed on extremely unbalanced data. Consequently, a variety of approaches have been specifically developed for outlier identification, some of which take into account not only the highly unbalanced scenario, but also the absence of ground truth class labels (unsupervised scenario) and the propensity of outliers to hide their behavior deep inside a specific subset of dimensions (Lazarevic & Kumar, 2005; Filzmoser *et al.*, 2008; Zhang, 2013); these three characteristics, namely, highly unbalanced data, unsupervised setting and outliers hidden in lower dimensionality, are predominant in outlier detection real-world datasets.

The relatively high rate at which new outlier detection algorithms are proposed in the literature (Hodge & Austin, 2004), increasingly somehow mirrors, the huge existence of algorithms in the classification and clustering literature. This still skewed similarity between these fields is not merely a coincidence, as the diversity of domains and types of data in which they operate require a wide and diverse set of algorithms. In the next subsections, we explore the main outlier detection categorizations.

4.3.1 Assumptions about the data

The diverse sets of outlier detection algorithms available in the literature are based on specific and constrained definitions about what should be considered an outlier. These constraints rep-

resent both the strength and weakness of outlier detection algorithms. Despite the aspiration of creating a single algorithm capable of operating on any type of dataset (Domingos, 2016), no single current classifier is able to model the different peculiarities found in real-world data; in fact, outlier detection does not deviate from this reality. Zimek (Zimek *et al.*, 2013) observed that algorithms used in outlier identification showed variable behavior, depending on the complexity of the dataset and the capacity of the algorithm to model data; furthermore, Aggarwal (Aggarwal & Sathe, 2015) argues that the parameter settings of the algorithm, such as the number of nearest neighbors *(*k) and the size of the subsample used, are not immune to different data scenarios, and should be selected carefully depending on the application domain.

4.3.2 Application domain

Each domain in outlier detection has different data characteristics, with specific types of outliers hidden in the data. Accordingly, the selection of an outlier detector is completely application-dependent, e.g. An outlier detection algorithm based on linear regression will attempt to find outliers by detecting those observations that have the largest deviation from a linear pattern. However, if the unusual behavior of these observations is not visible in a linear scenario, but instead, is depicted as isolated points far from main clusters in the data, then the linearity bias of this algorithm will hinder the algorithm's detection rate. A better approach in this scenario would involve using an outlier detector based on distances or densities. Selecting the right type of algorithm for a dataset is crucial to improve the detection rate by decreasing the inherent bias of the algorithm. Examples of outlier detection domains include fraud detection, medical anomaly diagnosis, irregular image detection, textual anomaly classification, sensor and damage prevention, etc. For an exhaustive and comprehensive description of outlier detection domains, please refer to (Aggarwal, 2015; Chandola *et al.*, 2009)).

4.3.3 Availability of labeled data

Depending on the level of accessibility to labeled data, outlier detection algorithms can be segregated into three groups. The first group comprises those algorithms requiring labeled

data for both outliers or inliers; this group of algorithms is generally referred to as supervised. Ground truth class labels are usually used in the algorithm's training phase to build a model that aims to distinguish between classes. The second group of algorithms is known as semi-supervised (Das *et al.*, 2016) and consists of algorithms capable of operating on semi-labeled data; here, there is information about the ground truth labels for only one of the classes, and usually, the labels for the outliers are available and the normal class is completely unlabeled or instead is partly labeled but with some unidentified outliers. The last group consists of fully unsupervised algorithms(Breunig *et al.*, 2000); this group does not require any labels, and can thus operate without knowledge of the true categorization of both outliers and inliers. A canonical example of this type of algorithms is the Local Outlier Factor (LOF) (Breunig *et al.*, 2000).

Although using labeled data can boost the discrimination between interesting and uninteresting outliers, thus improving detection rate, in most cases, the novel nature of these observations and consequently the absence of labeled data prohibits the use of supervised approaches. Moreover, an unsupervised setting is one of the most interesting, common and difficult scenarios in outlier detection

4.3.4 Parameters required

Parametric approaches (Rousseeuw & Hubert, 2011; Barnett & Lewis, 1994) assume data following a specific statistical distribution; however, this assumption is constantly violated. In the outlier detection scenario the same outliers that the user is attempting to isolate can influence the distribution parameters such as the mean and standard deviation. In contrast non-parametric approaches(Knorr & Ng, 1997; Breunig *et al.*, 2000; Ramaswamy *et al.*, 2000; Kriegel *et al.*, 2008), being more suited to outlier detection, do not assume a pre-specified type of distribution. The most common types of these approaches are the distance-based and density-based methods(Breunig *et al.*, 2000). The former try to find global outliers, while the latter attempts to find local ones. Throughout this study we focus, on the most common, non-parametric scenario.

4.3.5 Type of output

Outlier detection algorithms generally produce results either in the form of scores or binary labels. A dual classification is a handy type of output as it classifies observations simply as outliers or non-outliers, thus providing the final user with a hypothetical set of deviant observations; however, a simple binary discrimination often lacks the information that a degree implicitly offers. Accordingly, a numeric output score, such as the one produced by NN-based approaches (e.g. LOF (Breunig *et al.*, 2000)), provides an interesting insight into the degree of divergence of each observation; such information allows an empirical determination of a threshold for segregating outliers from inliers.

Despite the useful information that an outlier score provides, in most real world scenarios, the final user will eventually require an unequivocal binary decision about the identity and number of outliers, such threshold can be determined using domain knowledge or with extreme value discrimination methods (Knorr & Ng, 1997). This threshold is nevertheless completely application-dependent, and increasing or decreasing it has the indirect effect of diminishing the number of misclassified outliers or inliers, respectively. Thus, a threshold is usually fixed avoiding the misclassification of outliers (low false negatives rate), while attempting seizing the bulk of these (high true positive rate), at the expenses of incorrectly classifying some inliers (increasing false positive rate). This trade-off between false negatives and false positives is best determined empirically; however, if such knowledge is absent extreme value methods can be used to establish it.

For the experimentation with different density measures in an ensemble setting, we will use the LOF algorithm as the base detector (as was done previously by Lazarevic (Lazarevic & Kumar, 2005)). Besides being an iconic algorithm in the outlier detection literature, its instability as a function of variations in data size, dimensionality and parameter settings makes it a favorable candidate, under perturbation of these conditions, for evaluation in an ensemble setting.

4.4 Outlier ensembles

Ensembles of outlier detectors have been proposed as a mechanism to improve the robustness and detection rate of a single algorithm (Rokach, 2009); in outlier detection, the ensemble literature is rather limited, with a few approaches formally recognized as ensembles(Aggarwal, 2013b). Nonetheless, the field is quite interesting due to the inherent critical nature of outliers. The scarcity of ensembles approaches for outlier detection can be justified by the same quirks that characterize the field: non-availability of ground truth labels, highly unbalanced datasets, and lastly, the propensity of outliers to hide their behavior in specific subsets of dimensions. Thus, ensemble approaches focus on the improvement of the detection rate and processing time, while considering these three issues. Aggarwal (Aggarwal, 2013b) noted that not all ensembles approaches for outlier detection are self-identified by their corresponding authors as ensembles; instead, they are simply presented as outlier detection algorithms. Formally recognizing and categorizing them as ensemble approaches will contribute to improve the categorization of the currently scattered literature.

An interesting and useful classification of outlier ensembles was implemented in Aggarwal (2013b) by considering two variants: the way in which the algorithms in the ensemble collaborate, and the mechanism used as a source of diversity in the ensemble (Kuncheva, 2003; Hsu & Srivastava, 2009; Windeatt, 2005; Brown *et al.*, 2005). The former, also known as component independence, refers to the reliance of components on previous iterations of the ensemble (Das *et al.*, 2016). An ensemble is thus categorized as independent if the execution of a component does not have influence in the parametrization and execution of the remaining members, being the set of results self-sufficient and directly comparable for their posterior aggregation, or as a sequential ensemble if previous executions are used to refine the parameters and/or subspaces with which the next algorithm operates. The latter, known as component type, is based on the mechanism used to induce diversity in the ensemble. This induction is usually done either by using different algorithms (model-centered ensemble) or by using variations in the search space (data-centered ensemble). A model-centered ensemble attempts to generate diversity with the use of different hypotheses regarding the true, although hidden, outlier behav-

ior. This set of diverse hypotheses can be generated using distinct types of detectors (Nguyen *et al.*, 2010), parameterizations, initializations, etc. Although a data-centered ensemble is also based on different hypotheses, it however pursues diversity, not by variations in the algorithm used, but by limiting the subsamples of observations and/or subspaces (Leckie, 2016; Müller *et al.*, 2011) which are available for the algorithm to analyze. A classical example of this is feature bagging (Lazarevic & Kumar, 2005), which feeds a single algorithm, in iterations, with random subspaces of the dataset. This feature bagging mechanism acts as a source of diversity, which consequently tends to improve detection rate.

An ensemble approach for outlier detection is generally implemented, as described in Aggarwal (2013b), in three stages: first, a model is created by a single outlier detection algorithm. This model represents an ensemble component, and is created iteratively using different types of algorithms, data subspaces, data subsamples, parameter settings, etc. Secondly, different outlier detection algorithms tend to produce scores whose scales and interpretation vary widely, and attempting to use these scores without normalization could bias the ensemble process towards algorithms that inherently tend to produce scores with a wider range of values. This variability over ranges is not exclusively an artifact resulting from the use of heterogeneous detectors; rather, it is also detectable on scores produced by the same algorithm, but operating over different subsamples and/or subspaces of data, and even with different parameterizations of the algorithm. Normalization will raise the scores to a comparable scale, easing their combination. Finally, the normalized scores obtained in the previous phase are used as individual hypotheses which are subsequently combined to produce a unified output.

Although combination functions adopting advanced functionalities, such as those assigning specific weights to each feature or ensemble component (Pasillas-Díaz & Ratté, 2016b), allow a deeper understanding of the outlying behavior of some observations in the data, simple combination functions, such as a straightforward average, can significantly improve the detection rate of a single detector due to the variance reduction effect of combining different assessments (Aggarwal & Sathe, 2015; Chandra *et al.*, 2006). The results of this approach can be generally, conservative, assuming that each score must have the same influence in terms

of determining the final outlier score. Conversely, a maximum combination approach has the potential advantage of emphasizing observations in which at least one ensemble component assigned a relatively larger outlier score. Nevertheless, the maximum combination approach can also be heavily influenced by noisy observations, which then causes a detection rate inferior to that of its individual components. This behavior is mainly expected in small datasets (Aggarwal & Sathe, 2015). Notwithstanding the somewhat prevalent use of combination functions such as the average and maximum, more specialized combination functions have been proposed in the literature (Pasillas-Díaz & Ratté, 2016b; Kriegel *et al.*, 2011), however their use remains application-dependent.

Beyond the selection of an appropriate combination function, an ensemble of outlier detectors faces two essential problems: high-dimensional data and the tendency of outliers to hide in lower dimensional subspaces. These two problems are inherently correlated, as outliers in higher dimensional datasets tend to reveal their outlying behavior only on a specif subset of dimensions. The notion of what constitutes high-dimensional data is time-dependent. Early outlier detection approaches (Knorr *et al.*, 2000) were focused on dimensionalities much lower than what is currently observed in current datasets. Some approaches for outlier detection are designed to deal with this high-dimensional scenario. The authors in Aggarwal & Yu (2001) proposed the use of evolutionary search algorithms to search for lower and sparse dimensional projections of data (Filzmoser *et al.*, 2008), being outlier observations predominantly located in such sparse regions. Also, state of the art approaches attempt to deal with the high-dimensional scenario by searching for outliers in random sets of dimensions (Lazarevic & Kumar, 2005), random samples of data (Zimek *et al.*, 2013) , selected subspaces (Keller *et al.*, 2012; Müller *et al.*, 2011) or in a combination of random sets of samples and dimensions (Pasillas-Díaz & Ratté, 2016a); the mechanisms in some of these approaches not only reduce the search space and detect outliers hidden in lower dimensional subsets, but also act as a essential source of diversity for the ensemble.

Throughout this work, we use feature bagging (Lazarevic & Kumar, 2005) as the ensemble approach in our experiments with different distance measures. Its straightforward implemen-

tation, its potential to improve the results of a single classifier and the attention that it has received in the literature were the main aspects considered in its selection. It is worth mentioning that feature bagging, as proposed by Lazarevic & Kumar (2005), depends on the selection between two combination functions, Cumulative sum and Breadth first, to be used in the last phase of the approach. The results of Lazareviv showed, in almost all scenarios, that Cumulative sum (simple average) outperforms Breadth first in terms of the Area Under the Curve (AUC) (Lazarevic & Kumar, 2005). Thus, throughout our study, we assume the use of feature bagging in conjunction with Cumulative sum as the combination function.

4.5 Diagnostic tools

Evaluating an unsupervised outlier detection algorithm is a challenging exercise. The main obstacles arise from the very nature of outliers, such as their remarkably low proportion when compared to that of normal observations, the absence of labeled data, a potential corruption of normal data with unidentified outliers, a lack of datasets specifically designed for the evaluation of outlier detection algorithms, etc. These obstacles are not solely present in the evaluation of a single outlier detector, but are handed down to the ensemble scenario.

Evaluation measures commonly used in the classification field, such as accuracy (Soares *et al.*, 2006; Huang & Ling, 2005), are difficult to adapt in an outlier detection scheme. The inherent bias of accuracy toward more or less balanced datasets hinders its viability as an evaluation measure in outlier detection, for example, a simplistic algorithm applied to an imbalanced dataset (one of the main characteristics of the outlier detection domain) with a very low proportion of outliers, could achieve an almost perfect, but misleading, accuracy by simply classifying all observations under the inlier class. However, this measure does not only fail to correctly evaluate the results of the algorithm, but more importantly, losing focus the outlying observations, which are indeed the main goal of an outlier detector.

Commonly used evaluation approaches in outlier detection are classified either as external or internal. The former comprise those measures that evaluate the final algorithm decisions using

ground truth labels. This means that although an outlier detector may be able to operate in an unsupervised scenario, its results are evaluated using knowledge about the true identity of outliers and inliers. A straightforward procedure, which is regularly used in the literature, consists in removing or simply ignoring the data labels, hence using unlabeled data to feed the outlier detection algorithm; Thus, ground truth labels are only used in the last phase for evaluation purposes. Commonly used external evaluation approaches include ROC curves (Bradley, 1997; Fawcett, 2004, 2006), Area Under the ROC Curve (AUC) and precision@n (Schubert *et al.*, 2012). The latter type consists of measures that do not use ground truth labels to evaluate the results of an outlier detector, and which are thus completely oriented toward an unsupervised setting. In outlier detection only one seminal work has covered this kind of evaluation measure (Marques *et al.*, 2015), and it is essentially oriented toward an ensemble setting. This scarcity of internal evaluation measures is mainly due to the complexity of evaluating an algorithm in the absence of ground truth and of the highly imbalanced scenario of outlier detection. Care must be taken in using internal validity measures as any misleading assumption regarding the identity of true outliers present in the data can introduce an unforeseen bias into the process, and negatively affect the detection rate of the algorithm.

External evaluation measures constitute the prevailing type of measures encountered in the outlier detection literature, being the most used ROC curves, the area under the ROC curve (AUC), precision-recall curves and finally precision@n. ROC and precision-recall are similar types of curves as both plot the true positive rate (Recall) in one of their axis; however, they vary in the information plotted in the remaining axis, while ROC curves plot the false positive rate, precision-recall curves plot precision (percentage of detected outliers which are indeed true outliers). Despite the similarities between these curves, ROC curves are more easy to read and understand, thus, they are widely used in the literature literature(Lazarevic & Kumar, 2005; Zimek *et al.*, 2013, 2014, 2012; Aggarwal & Sathe, 2015).

As noted in Aggarwal & Sathe (2015), one concern when evaluating the results of an outlier detector based on distances, resides in selecting a single number of neighbors (k) for different data sizes. citepAggarwal2015t states that “In data sets, where the accuracy of a k-NN algo-

rithm increases with k on the full data set, subsampling with fixed k will generally improve the accuracy of an individual detector on a single subsample"; thus, failing to adjust k to a specific subsample makes the bias component dependent on the parameter and sample size selected. However, distinct approaches in literature (Campos *et al.*, 2015; Zimek *et al.*, 2013), including iconic approaches like feature bagging (Lazarevic & Kumar, 2005), are still based on experiments with a fixed k. Accordingly, in our experiments we considered both schemes of fixing or adjusting k.

4.6 Evaluation

4.6.1 Methods

As mentioned in Sections 4.3 and 4.4, LOF and feature bagging, respectively a single outlier detection algorithm and an ensemble approach, are two iconic and extensively used approaches in the outlier detection literature. Accordingly, we based our experiments on these two approaches. LOF and feature bagging require that a couple of parameters be specified. The former depends upon the selection of the nearest neighbors (k), while the latter requires the specification of number of iterations or ensemble components. Unless explicitly specified (as in our experiments with real-world data), we set k to 5. For feature bagging we fixed the number of iterations to 10. Although these two parameters were fixed in most of our experiments, it is important to note that two exception were the experiments on *Synthetic_batch03* and those on real-world datasets, where these parameters were adjusted to provide a deeper and richer set of scenarios.

4.6.2 Datasets

Outlier detection algorithms are generally evaluated using synthetically created datasets or datasets originating from the classification field (Lazarevic & Kumar, 2005; Zimek *et al.*, 2013; Kriegel *et al.*, 2011), but adapted to the outlier detection scenario. This practice originated from a lack of datasets specifically designed for outlier detection. The adaptation process is simpler

Table 4.1 Datasets characteristics

Datasets	Classes	Attributes	Noisy Attributes	Observations	Outliers	Subsampled Outliers%	Dataset Adjustments
Synthetic_batch1	2	10	0	10-500	1-5	1%	—
Synthetic_batch2	2	5-400	0	500	5	1%	—
Synthetic_batch3	2	0	0	0	0	1%	—
Breast cancer	2	32	—	569	21	3.56%	Class 2 v/s. 1
Ionosphere	2	34	—	237	12	5.06%	Class 1 v/s. 0
Lymphography	4	18	—	148	6	3.90%	Merged class 1 & 4 v/s. rest
Satimage	7	36	—	6435	62 - 70	1.06 - 2.5%	Class 2,4 || 5 v/s rest
Shuttle	7	9	—	14500	2 - 39	0.02 - 0.34%	Class 2,3,6 || 7 v/s rest
Waveform	3	21	—	3343	164 - 169	4.66 - 4.87%	Each class v/s. the rest

when the dataset already has a minority class, which is selected as the set of outliers. However, in datasets without an obvious minority set of observations, a randomly selected class is down-sampled to represent the set of outliers. As noted in Campos *et al.* (2015) this mechanism of adapting datasets to the outlier detection scenario can inherently hinder the evaluation process of an algorithm by measuring its outlier detection rate in an artificial minority class, which can or cannot represent a true deviant observation. A better mechanism for evaluating outlier detection algorithms could involve the use of internal validation measures, which can theoretically assess the performance of an outlier detection algorithm without using labels; however, in the outlier detection literature, only a few seminal works have proceeded in this direction (Marques *et al.*, 2015), and these tend to be computationally expensive. As mentioned in Section 4.5, we used the conventional and straightforward external evaluation procedure.

Thus, regarding experimentation with different distance measures and the impact on the detection rate and complexity time, we created or adapted distinct sets of synthetic and real-world datasets (Table 4.1). In the following subsections 4.6.2.1 and 4.6.2.2 we detail the specific processes followed for each dataset.

4.6.2.1 Synthetic datasets

In the interest of evaluating the behavior of an outlier detector when controlling internal and external factors such as data size and dimensionality or the specific parameters of the algorithm, we generated 3 different synthetic batches of datasets (Table 4.1). Each scenario was generated with the purpose of evaluating a specific data or algorithm perspective. In all three cases, the percentage of outliers was set to 1%, irrespective of the dimensionality of the data.

- *Synthetic_batch01*. The first batch of synthetic datasets was generated with only 10 dimensions, and the number of observations was varied between 500 and 10000 (with sequential increments of 500 observations per iteration). Using this method, we generated 40 datasets for *Synthetic_batch01*. This batch of datasets was intended to evaluate the impact of a dis-

tance measure on the performance and detection rate of a single detector and ensembles of detectors when varying the size of the data.

- *Synthetic_batch02*. The second batch of datasets was generated by setting the number of observations to 500, and varying the dimensionality of the data between 10 and 400 attributes. Using this mechanism we generated 40 datasets. This batch was intended to evaluate the impact of different distance measures on the detection rate and performance of a detector and groups of detectors as the data dimensionality is increased.

- *Synthetic_batch03*. The final batch of datasets was generated by setting the number of observations to 500 and the quantity of dimensions to 10. This batch consisted of a single dataset. The purpose of the batch was to measure the detection rate and processing time of an ensemble of detectors as the number of ensemble components increased while, keeping the data size and dimensionality fixed.

4.6.2.2 Real-world datasets

We used different sets of synthetic datasets to measure the behavior of an outlier detector under controlled data conditions. However, limiting our experiments exclusively to synthetic datasets would also limit our potential for further exploration of the detectors when facing real-world environments. Real-world data provides a richer set of conditions not limited to those pre-established in synthetic datasets. We selected six real-world datasets from the UCI machine learning repository (Bache & Lichman, 2013). We followed the same procedure as set out in Lazarevic & Kumar (2005) to adapt some of the datasets to the binary, unsupervised and unbalanced outlier detection scenario, this mechanism to adapt classification datasets to the outlier detection task is the prevalent procedure used in the outlier detection literature (Lazarevic & Kumar, 2005; Zimek *et al.*, 2013; Kriegel *et al.*, 2011). The procedure consists roughly in selecting the observations in the minority class, if present, to act as outliers. If needed, down-sampling can be used to further decrease the proportion of outliers. In datasets where

there is not an obvious minority class, one of the classes is downsampled and used as the outlier class, and the remaining observations are then merged and used as inliers.

The specific adaptations performed on each dataset can be seen in Table 4.1. The *Breast cancer* and *Ionosphere* datasets already had two classes. The classes of the former are malignant and benign, with malignant being the minority class. In the latter dataset, the classes are good or bad, depending on whether there is some structure in the ionosphere or whether there is no structure (which allows some signals to pass through it), in this scenario we used the latter case as the minority class. In both datasets, we downsampled the minority class to 10%, and used it as the outlier class. In the *Lymphography* dataset with four classes (normal find, metastases, malign lymph and fibrosis), we merged the first and fourth classes to act as the outlier class, and the remaining classes were used as inliers. The *Satimage*, *Shuttle* and *Waveform* datasets had more than one minority class whose observations could be used as outliers. *Satimage* consisted of satellite images (multi-spectral values of pixels) classified into seven classes corresponding to different types of soils, we identified classes two, four and five as the minority classes. Similarly, the *Shuttle* dataset consisted of seven classes, and we observed the relatively small proportion of observations in classes two, tree, six and seven. In the previous two datasets, *Satimage* and *Shuttle*, we used each minority class, in turns, to act as the outlier class. For the *Waveform* dataset, which consisted of three classes of waveforms (each class with a similar number of observations), we used each of the classes to act, in turns, as the set of outlying observations. With this procedure of adapting datasets to the outlier detection scenario, we obtained 13 datasets based on real-world data.

4.6.3 Results

On evaluation of the impact that distinct distance measures have on the performance and detection rate of an outlier detector or an ensemble of detectors, LOF and feature bagging, respectively, we explored their behaviors under different settings, using synthetic and real-world datasets. First, we examined the impact that size (*Synthetic_batch01*) and dimensionality (*Synthetic_batch02*) had on detection rate and processing time when using a single classifier or an

ensemble approach. Then, we explored how an ensemble is affected, in terms of detection rate and processing time, by the distance measure used (*Synthetic_batch03*), moreover, using such batch of datasets, we also examined the behavior of the distance metrics when interacting with distinct ensemble sizes. Finally, we experimented with different values of k for LOF and feature bagging, using real-world datasets commonly used in the outlier detection literature (Table 4.1).

In all the scenarios, being based either on synthetic or real world-data, we performed the experiments 10 times and averaged the results. This mechanism was used in order to reduce the variability, in processing time and scores, that a single run of an algorithm can exhibit, due to the specific dataset or subsample used. This is particularly true in the case of synthetic datasets, where such fluctuations originated from the implicit randomness of the mechanism used to generate the data (e.g. the algorithm could be fed by chance, with an easy or hard dataset, thus generating misguiding positive or negative results, respectively, further masking the expected real performance of the algorithm). In our experiments with real-world datasets, such randomness in the generation process is in general not explicitly considered (although we acknowledge that such randomness can also be present as the available data could also represent only a sample of a finite, but unknown, set of data). In addition to the data variability, we further considered the variability due to the implicit random mechanisms in the algorithms used in our experiments. Feature bagging uses two random mechanisms, first to set the quantity of features to be sampled, and second, for the specific random features to be used in each iteration of the algorithm. As in the case of synthetic data, these two random mechanisms produced variable results. An exception was with our real-world data experiments with LOF, where invariably, the algorithm analyzed the same data sample. Accordingly, basing our experiments on different iterations allowed us to have more consistent results and to expose a trend in the behavior of the algorithms.

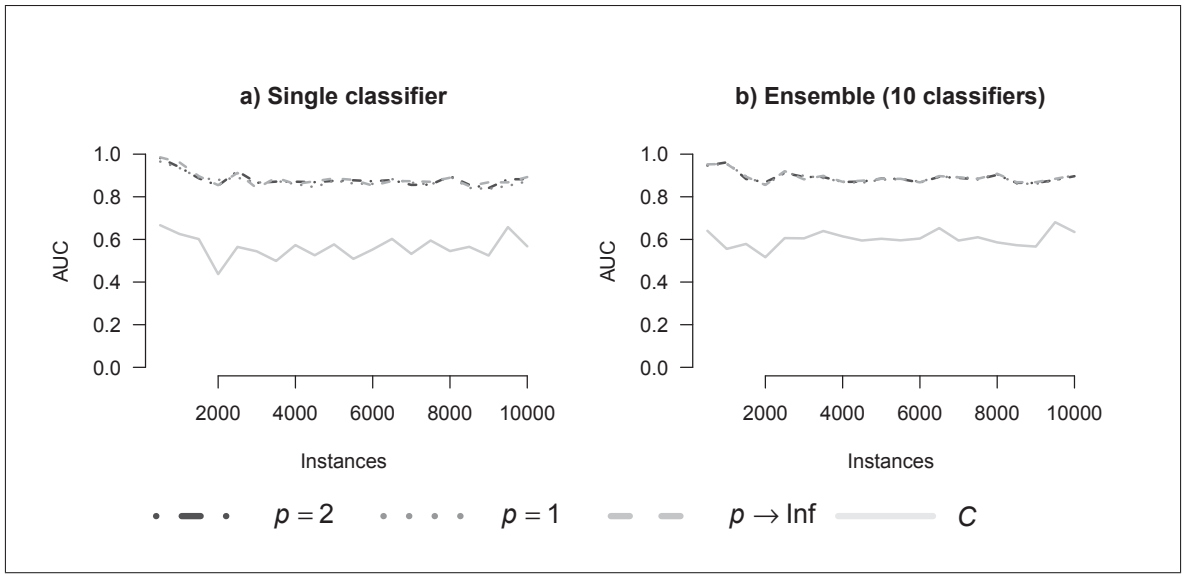


Figure 4.1 AUC with an increasing number of instances for LOF (left) and Feature bagging (right) on Synthetic_batch01.p=2 Euclidean, p=1 Manhattan, $p \rightarrow \infty$ Chebyshev, C Canberra.

4.6.3.1 Synthetic data

In the experiments with *Synthetic_batch01*, we evaluated the impact of distinct distance measures on the detection rate and processing time as data size increased. Under this scenario, we observed a similar behavior for Minkowski p=1, p=2 and $p \rightarrow \infty$ (Figure 4.1(a)); an exception was Canberra which, remarkably, had the lowest AUC. Our experiments on an equivalent data scenario, but using an ensemble of detectors, showed similar results to those obtained by a single classifier (Figure 4.1(b)). Despite the similarity of the results obtained in both cases, the ensemble approach slightly smoothed the variability noticed in our results with a single classifier, this effect was more pronounced for the Canberra metric. It is worth to mention that the variability found in a single detector was already smoothed due the mechanism used to reduce the disparity of results due to randomness in the generation of our synthetic scenarios (Section 4.6.2).

Our processing time assessment of *Synthetic_batch01* showed a similar tendency for all distance metrics (Figures 4.2(a), 4.2(c)). However, on closer examination, an increasing discrep-

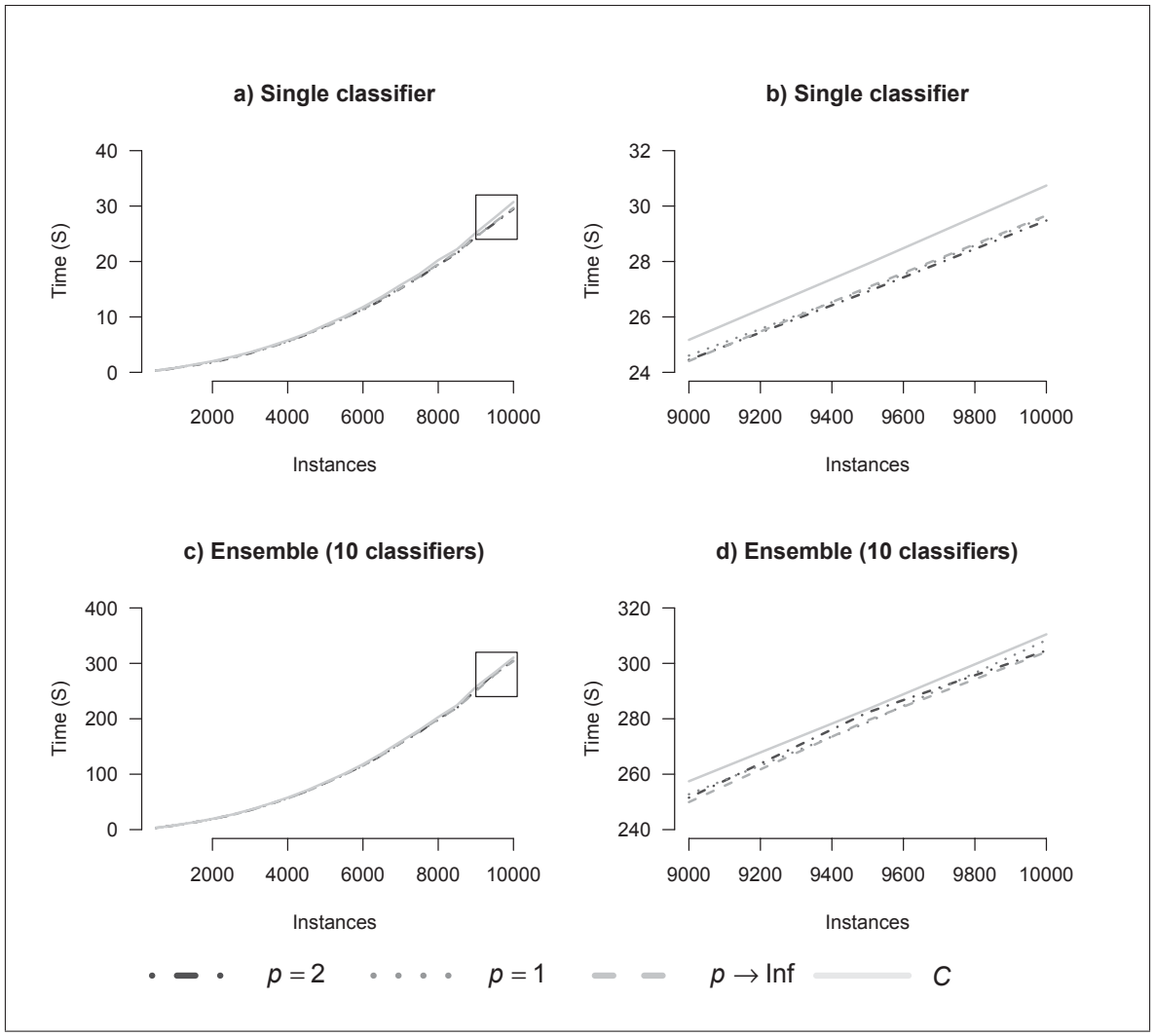


Figure 4.2 Time with an increasing number of instances for LOF ((a) and (b)) and feature bagging ((c) and (d)) on Synthetic_batch01. p=2 Euclidean, p=1 Manhattan, $p \rightarrow \infty$ Chebyshev, C Canberra.

ancy was seen between Canberra and the others metrics (Figures 4.2(b), 4.2(d)); the hardness of Canberra in *Synthetic_batch01* related to AUC was also exhibited in its processing time, bearing the highest computational cost among all distance metrics. It is worth mentioning that this discrepancy was almost indistinguishable for a dataset with less than 6,000 observations (Figures 4.2(a), 4.2(c)), and it was not until the number of observations was increased beyond 9,000 that a gap in the processing times appeared between Canberra and the remaining metrics (Figures 4.2(b), 4.2(d)). Overall, LOF and feature bagging processing times increased steadily.

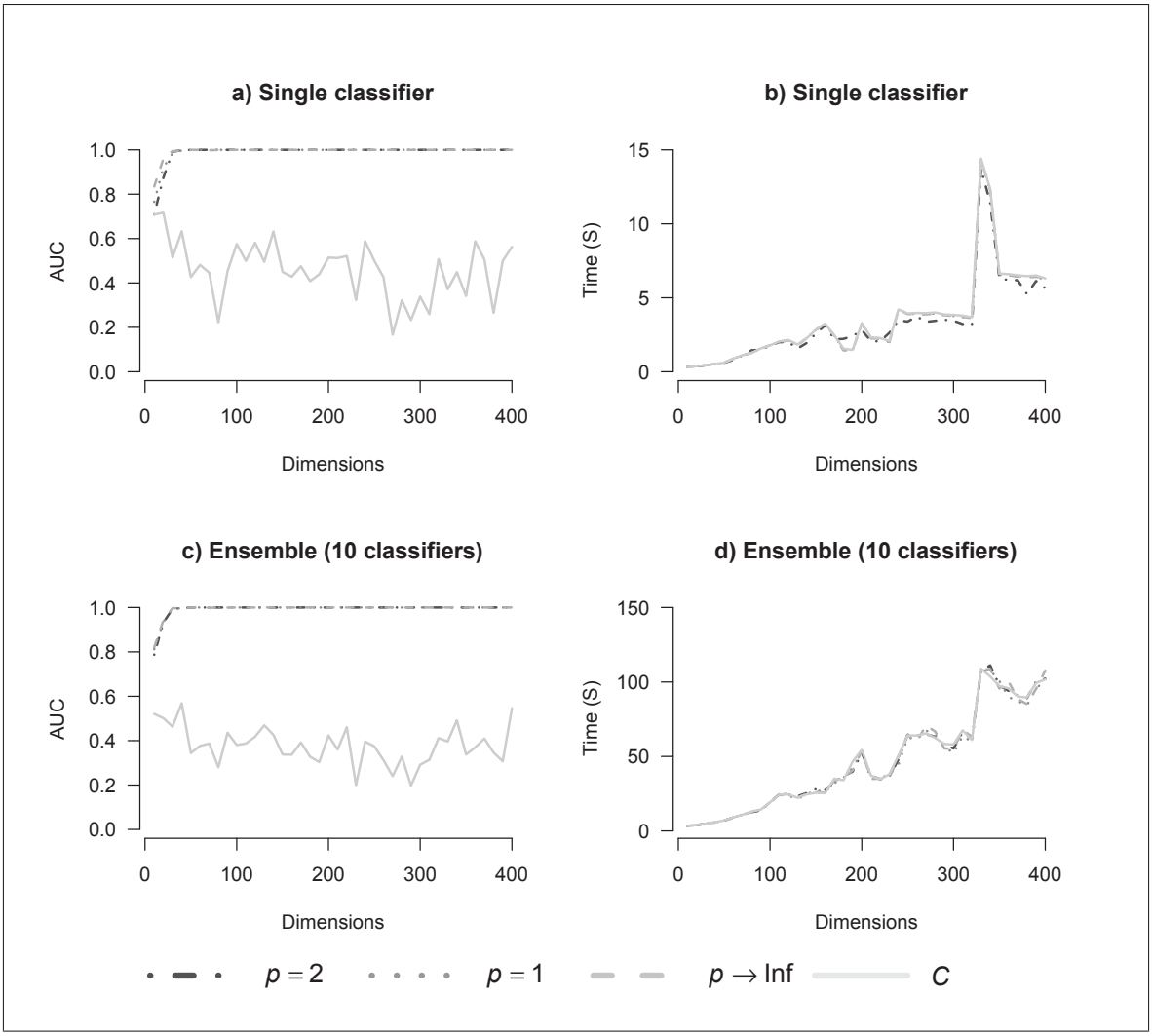


Figure 4.3 AUC (left) and time (right) for LOF ((a) and (b)) and feature bagging((c) and (d)) with an increasing number of dimensions on Synthetic_batch02. p=2 Euclidean, p=1 Manhattan, $p \to \infty$ Chebyshev, C Canberra.

Such a behavior was expected due to the intrinsic sensitivity of LOF, which is also the base algorithm in feature bagging, to the number of observations n, and a complexity of $\mathcal{O}(n^2)$.

The processing time and detection rate of LOF, and thus that of feature bagging, depends not only on the number of observations, but also, it depends heavily on the dimensionality of the data; accordingly, we used *Synthetic_batch02* to evaluate these characteristics while varying the distance metric used. In our experiments with LOF in *Synthetic_batch02*, there was a similar AUC trend with the Minkowski distances, with p=1, p=2 and $p \to \infty$ (Figure 4.3(a)).

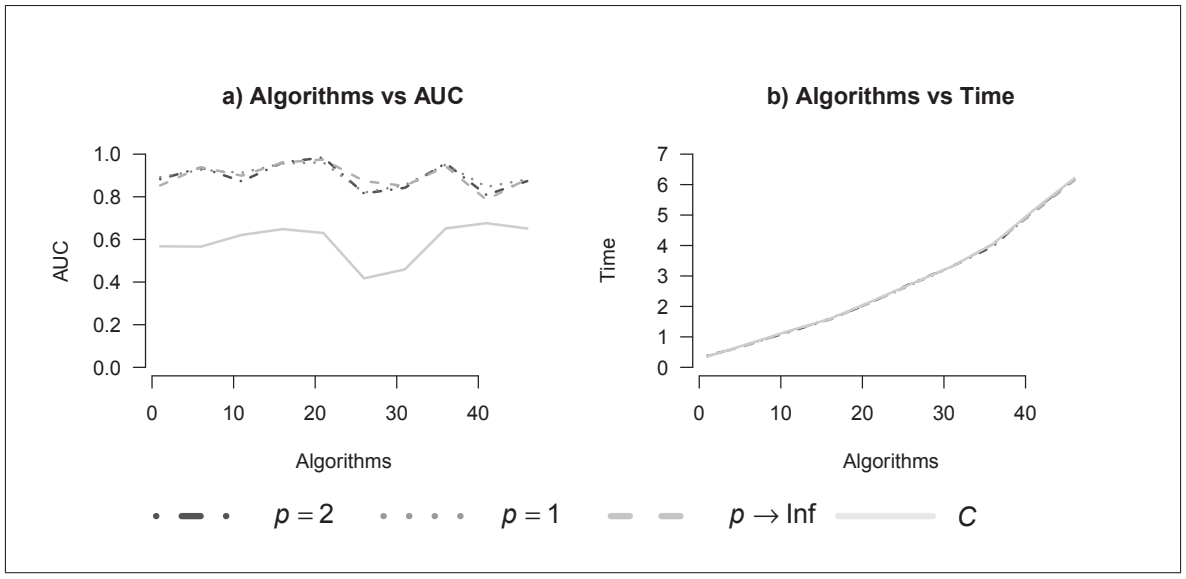


Figure 4.4 AUC(a) and Time(b) for FB with an increasing number of algorithms on Synthetic_batch03. p=2 Euclidean, p=1 Manhattan, $p \to \infty$ Chebyshev, C Canberra.

The exception was the Canberra distance, which showed the lowest AUC amongst all measures. A similar behavior was observed in feature bagging (Figure 4.3(c)); however, the fluctuation in AUC observed in LOF (Figure 4.3(a)) was smoothed by feature bagging (Figure 4.3(c)). The Canberra measure not only showed the lowest AUC, but also, its processing time was slightly higher than that of Euclidean, Manhattan and Chebyshev (Figures 4.3(b), 4.3(d)), although the difference in processing time was not as pronounced as that in AUC, and was further reduced in an ensemble setting.

Up to this point, our experiments on synthetic data evaluated the effects that a specific distance measure had on detection rate and processing time, considering factors such as the size and dimensionality of the data; however, the performance of an outlier detector, specifically, an ensemble of classifiers, is also affected by the number of ensemble components. Accordingly, we performed experiments on *Synthetic_batch03* with a static data size and dimensionality. These experiments consisted of 10 independent iterations of feature bagging, with the number of components increased from 1 to 46, in increments of 5. Invariably, an ensemble with a single component simply represents a sole iteration of LOF over a set of randomly selected features. Our experiments on *Synthetic_batch03* revealed that small, but highly variable, increments in

AUC can be achieved, for all distance metrics, by increasing the number of ensemble components beyond 1 (Figure 4.4(a)); however, this improvement was not constant, and apparently ceased when the number of components was increased beyond 20. Minkowski with p=1, p=2 and $p \to \infty$ showed a similar behavior; the exception was Canberra, which invariably exhibited the lowest AUC. Furthermore, our experiments showed, for all distance metrics, a constant and practically indistinguishable increments in processing time as the number of ensemble components increased (Figure 4.4(b)).

4.6.3.2 Real-world data

The sets of artificially created data provided three different scenarios which allowed the evaluation of distinct combinations of parameter settings, data and distance metrics. Such synthetic datasets allowed us to regulate the quantity of outliers, the presence or absence of noisy attributes, the number of observations, the dimensionality of the data, etc. However, beyond such appealing characteristics, because of the mechanisms used in their generation, artificially created datasets could inherently and probably inadvertently, be biased towards specif data structures more or less favorable to a particular distance metric, parameter setting, etc.; moreover, synthetic data also lacks some of the characteristics found in real scenarios, such as an unknown quantity and identity of noisy dimensions, the variable nature of outliers, the presence of noisy observations, a fluctuating proportion of anomalies, etc. Consequently, with the main aim of providing a richer set of evaluation scenarios, we performed further experiments on real-world datasets (Table 4.1). Such experiments using real world-data are common in the outlier detection literature. We selected six previously used datasets (Lazarevic & Kumar, 2005; Pasillas-Díaz & Ratté, 2016a; Zimek *et al.*, 2013). Such collections of data exhibit different particularities in terms of size, dimensionality, proportion of outliers, and more important, the specific differences expected due to the domain of origin. We followed the procedures detailed in subsection 4.6.2.2 to adapt classification-related datasets to the outlier detection scenario. Following this procedure, we generated a total of 13 datasets.

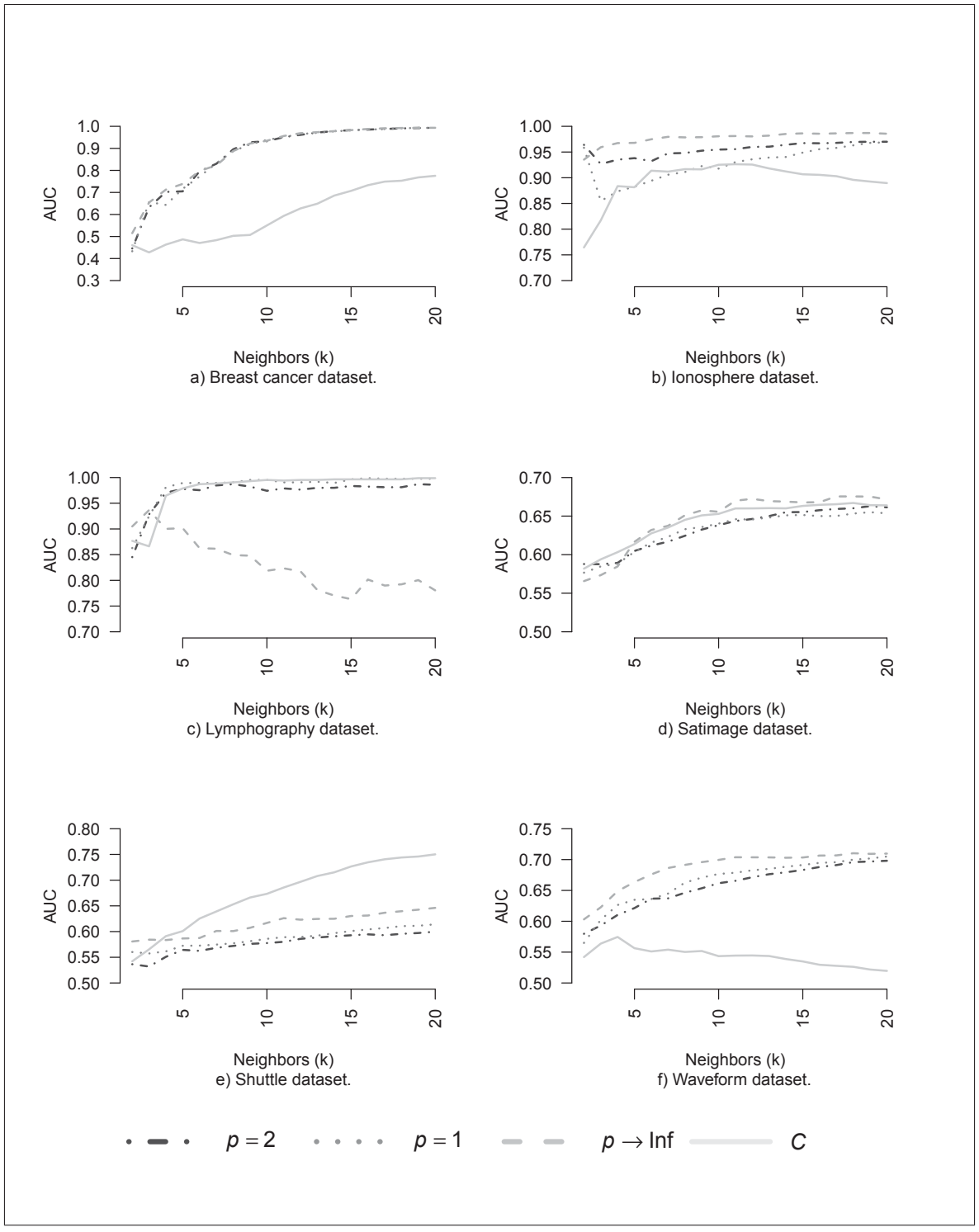


Figure 4.5 AUC for LOF, neighbors k=2 : 20, on real world datasets datasets. p=2 Euclidean, p=1 Manhattan, $p \to \infty$ Chebyshev, C Canberra.

As mentioned in Section 4.6.1, LOF and feature bagging (with LOF as its base algorithm) are intrinsically based on the computation of distances and densities between the k neighbors of each observation D_i in the dataset D. Accordingly, beyond a passive distance metric-dataset evaluation, we foresaw a series of experiments in which reach-real world datasets are iteratively examined, both by a single detector and an ensemble of detectors, with different values of k. Following this procedure, we evaluated the 13 adapted real-world datasets (Table 4.1). We iteratively incremented the number of neighbors from 2 to 20, with increments of 1. Thus, we performed 19 experiments with each of the datasets and averaged the results. This set of experiments allowed us to portray the behavior of a single outlier detector and an ensemble of detectors from the perspective of the interaction neighbors - distance metric. For ease of display, we separated the results obtained with a single classifier (Figure 4.5) from those attained with an ensemble of detectors (Figure 4.6). It is worth noting that we intentionally omitted, in this set of experiments, the figures related to processing time, as they exhibited a similar behavior to that observed in artificially generated datasets, and thus do not provided any additional information in this aspect.

Contrasting our experiments using a single detector with those based on an ensemble approach, we observed that, as expected , in most of the cases, feature bagging tended to improve the AUC of LOF. However, on observation of Figures 4.5, 4.6, see two main peculiarities: an inconstant behavior of distance metrics through different datasets and the high disparity in feature bagging improvements.

Diverging from our experiments with synthetic data, which constantly showed Canberra as the metric with the poorest detection rate, irrespective of size or dimensionality of the data, or the number of ensemble components, experiments with real-world datasets showed a different perspective. Contrasting the plots in Figure 4.5 and 4.6 we noted that not single distance metrics showed a consistently poor or superior behavior, even the Canberra metric showed a remarkably large AUC in the Lymphography, Satimage and Shuttle datasets; however, despite its exceptional detection rate in 3 datasets, it also exhibited the worst detection rate in the Breast cancer, Ionosphere and Waveform datasets (Figures 4.5(a), 4.5(b), 4.5(f)). We observed

in Figure 4.5(c) that Minkowski with $p \to \infty$ showed a remarkably poor detection rate in the Lymphography dataset, which further worsened as k increased. Contrasting the results in Figure 4.5(c) and Figure 4.6(c), we observed that the poor detection rate of Minkowski with $p \to \infty$ was lessened with the use of feature bagging; however, despite such improvements, its AUC continued to be negatively affected by increments in k, albeit inconstantly. Similar improvements, although less pronounced, were observed in all the sets of real world data. However, such gains in AUC are distinct for each datasets and dependent on the AUC achieved by the distance metric used by the base algorithm.

Feature bagging, as stated by its authors (Lazarevic & Kumar, 2005), certainly improved the AUC of LOF in most of the cases; however, this improvement was highly inconstant, and was even disadvantageous in noticeable combinations of distance metric and k (e.g. Figure 4.6(e) Canberra metric). We hypothesize on the mechanism behind this seemingly aberrant behavior in Section 4.6.4. Our experiments showed a tendency in feature bagging to show larger improvements for distinct aggregations of LOF and distance metric, mainly in distance metrics which tended to produce AUCs remarkably lower than the rest of the metrics. We observed that this effect was more significant in the Lymphography dataset for Minkowski with $p \to \infty$ (Figures 4.5(c), 4.6(c)). We observed a similar effect in the Waveform dataset, where Canberra, the metric with the worst AUC in the LOF case (Figure 4.5(f)), exhibited the larger improvement across all distinct measures (Figure 4.6(f)).

In addition to the two previous peculiarities, namely, inconstant behavior of the distance metrics and disparity in the improvements provided by feature bagging, our results, as displayed in Figures 4.5, 4.6, further showed that a more stable and consistent behavior can be attained by using basic Minkowski metrics like Euclidean (p=2) or Manhattan (p=1); however, such stability in performance remained generally conservative, being superior to the other metrics only in the Breast cancer dataset (Figures 4.5(a), 4.6(a)).

4.6.4 Discussion

In Section 4.6.3 we depicted a set of experiments directed towards the evaluation of an outlier detector in its intersection with distinct distance metrics and different data characteristics. This evaluation was performed on a diverse set of synthetic and real-world data scenarios, by altering the characteristics of the data, such as its size and dimensionality, or by adjusting some parameters of the algorithm like the quantity of ensemble components and the number of nearest neighbors.

We observed that across all distance metrics the processing time for a single outlier detector and ensemble of detectors in *Synthetic_batch01* was practically indistinguishable when the quantity of data observations was lower than 9,000 (Figures 4.2(a), 4.2(c)); however, an increasing difference in processing times between Canberra and the remaining metrics appeared as the size of the data exceeded 9,000 observations. With Canberra being a weighted version of the Manhattan distance, its processing time is largely affected by the weight factor in the denominator of Equation 4.4, this is evident on higher data sizes (Figures 4.2(b), 4.2(d)). Interestingly, it seems that its impact was almost insignificant in smaller datasets, however, its share in the processing time of the algorithms increased steadily with data size.

Overall, the processing time of LOF ($\mathcal{O}(n^2)$) is codependent on two factors: data size and dimensionality, while the number of observations (n) directly affects and practically dominates the processing time needed to compute the distance between all observations (Figure 4.2), its dimensionality or number attributes further hinders the processing time of LOF, as dimensionality increases so does processing time (Figures 4.3(b)). Moreover, like most ensemble approaches, feature bagging, besides the impact of size and dimensionality in processing time (Figure 4.3(d)), is also affected by the number of ensemble components (T). Thus, its processing time increases $\mathcal{O}(n^2 * T)$ as additional components are added to the ensemble (Figure 4.4(b)). Therefore, the processing time of an ensemble of outlier detectors like feature bagging was considered to be affected by 3 factors: data size, dimensionality and the number of components. However, the processing time of an ensemble of distance-based algorithms is further

affected by a fourth factor, namely the distance metric used to compute the distance between observations. State-of-the-art mechanism can be used to counteract, or at least lessen, the effect of the first two factors on the processing time. Such approaches sample the data (Zimek *et al.*, 2013) , the dimensions (Lazarevic & Kumar, 2005), or both (Pasillas-Díaz & Ratté, 2016a), while other approaches attempt to select a subset of relevant subspaces (Keller *et al.*, 2012; Müller *et al.*, 2011) or search for outliers in a transformed projections of the data (Filzmoser *et al.*, 2008). The ability of these approaches to reduce processing time is only a welcome side effect of their mechanisms used to induce diversity in the ensemble (thus decreasing variance and increasing detection rate) by computing outlier scores based on different instantiations of the data.

We expect that our results with different distance metrics can be generalized to current and future approaches for unsupervised outlier detection, moreover to an ensemble setting irrespective of the method used to induce diversity. Accordingly, based in our experiments, despite the similitude in processing time of the four distance metrics in small datasets, a weighted metric, like Canberra, should exhibit the highest processing time in relatively large datasets, independently of the ensemble approach used. Thus, in an outlier detection scenario where the time of execution is the main concern, a unweighted Minkowski metric like Manhattan (p=1) should be used. However, despite the potential concern on execution time, the detection rate of an algorithm continues to be the main target in outlier detection. In the following paragraphs, we examine the impact that different distance metrics have on AUC.

The execution time and detection rate of an algorithm constitute a critical trade-off in outlier detection. Usually, mainly in an ensemble setting, the higher the processing time, the higher the detection rate. However, our experiments on synthetic data revealed that there are special cases where a higher processing time can also be accompanied by a low detection rate. Canberra metric, which showed a similar execution time when compared with the set of Euclidean metrics, also exhibited the highest processing time when the number of observations was increased beyond 9,000 (Figure 4.2); thus besides having the highest processing time, Canberra also exhibited the lowest detection rate in synthetic scenarios *Synthetic_batch01* (Figure

4.1(a)) and *Synthetic_batch03* (Figure 4.4), similar results were observed in our experiments in an ensemble setting in *Synthetic_batch01*. Our experiments with *Synthetic_batch02* also exhibited Canberra as the metric with the worst detection rate (Figures 4.3(a), 4.3(c)), but in this scenario, Canberra showed a similar execution time as the remaining metrics (Figures 4.3(b), 4.3(d)). This tendency of Canberra to show a consistently lower detection rate in our synthetically created datasets was not, or at least was not consistently, observable in real-world data (Figures 4.5, 4.6). Real-world data seems to provide richer scenarios than those provided by our synthetic datasets.

On experimentation with synthetic data, we contemplate an implicit variability in AUCs due to the intersection of randomness in the data generation process and the use of a single detector. Accordingly, we reduced this variability with the averaging procedure described in Section 4.6.3.1. This produced results in which the effects related to the data generation process were, if not eliminated, at least lessened. Partially isolating this variability allowed us to depict a reduction in variability due purely to the use of an ensemble of detectors with a specific distance metric (Figure 4.1 (a) and (b)). Differently to the reduction due to the data generation process in the synthetic datasets, this reduction was essentially afforded by the ensemble, and this was confirmed in our results with real-world data (Figure 4.6.3.2), where there was no explicit randomness in the data generating process, being such reduction purely algorithmically driven.

Ensemble approaches for outlier detection, like feature bagging, which are based on the combination of multiple hypothesis about the outlier behavior of each observation in the data, clearly provide a reduction in the variability of results when compared with those produced by a single detector. Such ability of an ensemble of detectors is not new and has been studied and used previously in the literature; however, on examination of the AUCs generated with a single detector and those achieved with an ensemble of detectors (Figures 4.5, 4.6), we observed, as mentioned in Section 4.6.3.2, two peculiarities in the behavior of feature bagging: an inconsistent tendency in its results and a variability in the gains over a single classifier. Having partially isolated the variability due to the data random generation process, we argue that this seemingly inconsistent and variable behavior is due to the randomness in the processes used by

feature bagging and also by the ability of ensemble approaches to exhibit larger improvements with a base algorithm whose detection rate is slightly above that of random guess. Thus, a distance metric in a base algorithm exhibiting a modest performance, showed the most interesting improvements in detection rate, on the contrary if the individual performance of the base algorithm and distance metric leaves little room for improvement, then the gains provided by feature bagging are limited.

Our results in *Synthetic_batch01*, in pair with similar studies, showed that feature bagging certainly provides an improvement in detection rate; however, this was modest and variable. This small improvement in the AUC can be explained by the already high AUC achieved by LOF. The cases where feature bagging provided the largest gains were those scenarios where LOF performed particularly poorly; moreover, they were subject and limited to the individual detection rate achieved by each component and distance metric. As has been previously stated in the outlier detection literature, the range in the improvements provided by feature bagging is a function of the detection rates of its base algorithms, showing larger improvements in AUC when the detection rate of its base component is slightly above that of a random guess (e.g., AUC >0.5). In our study, such improvements were more notorious in real-world data, whereas they were minor in our synthetic scenarios. In our set of synthetic scenarios, LOF had already achieved a relatively high AUC, leaving feature bagging with little room for improvement. Nonetheless, even in our synthetic scenarios, an ensemble of detectors seemed to provide a stabilizing effect on AUC, which was better appreciated for distance metrics with a low AUC, such as Canberra (Figures 4.1(a), 4.1(b) & 4.3(a), 4.3(c)). Such stabilizing behavior could also be seen in our experiments with an increasing number of dimensions, an ensemble setting in *Synthetic_batch02* provided a more stable set of results when compared with those achieved by its individual component. This stabilizing effect was prominent for the Canberra metric (Figures 4.3(a), 4.3(b)). Interestingly, such a stabilizing behavior was also extended to processing time (Figures 4.3(b), 4.3(d)).

Independently of the distance metric used by its base component, feature bagging showed a variability in its results, mainly due to its two random internal processes, first to randomly de-

termine the number of dimensions to be used in the different iterations T_i of the ensemble T, and second to randomly assign the specific sets of dimensions to be used by each T_i. Notwithstanding the variability in the improvements provided by feature bagging, specific distance metrics, like Canberra, showed the largest improvements in detection rate, Canberra tend to exhibit at completely distinct behavior depending on the dataset under study. This effect was mainly, or least strongly, observed in Chebyshev ($p \to \infty$) in the lymphogrpahy dataset (Figures 4.5(c), 4.6(c)) and in Canberra in the waveform datasets (Figures 4.5(f), 4.6(f)). The largest improvements are provided by specific combinations of base component, distance metric and dataset, which produce results with a modest and variable detection rate. Thus, feature bagging, or any ensemble approach oriented towards variance reduction, could be used to stabilize and improve the detection rate of, otherwise, unstable distance metrics.

Overall, the selection and parametrization of an algorithm in the interaction outlier detector - distance metric - data, is primarily influenced by the trade-off between detection rate and execution time. Although a slow algorithm would be unacceptable in most domains where outlier detection can operate, the detection rate continues to be the main concern in most scenarios. Accordingly, in this study we attempted to provide a mechanism to select a distance metric not merely by blindly selecting the fastest or most accurate metric, but instead by guiding in the intricate combination of detector - parametrization - data, using distinct metrics, data sizes, dimensionalities, number of ensemble components, parametrization of the base algorithms, etc.

Notwithstanding our explicit attempt to provide a rich and complete set of evaluation scenarios, we acknowledge that our study does not exhaustively contemplated all the array of characteristics in the outlier detection scheme that could possibly affect the detection rate and processing time of an algorithm based on a specific distance metric. This evaluation was performed using different synthetic and real-world data scenarios. While synthetic scenarios allowed the evaluation of each distance metric over different data and algorithmic characteristics, real-world data unveiled an interesting behavior for unweighted Minkowski metrics. Euclidean distance has generally been used as the default metric for algorithms whose main mechanism depends on the computation of distances; in fact, even most of the libraries available for data analysis

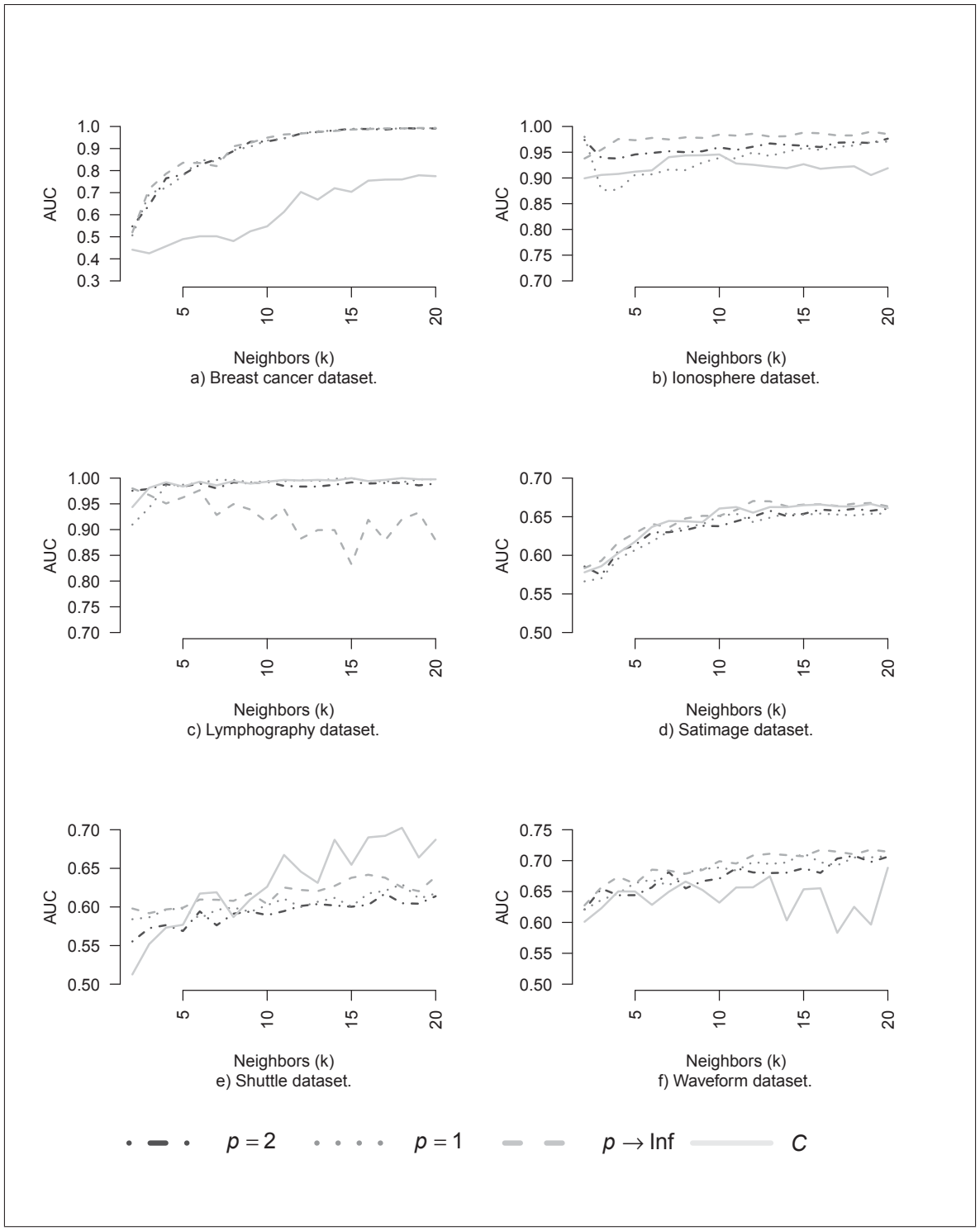


Figure 4.6 AUC for Feature bagging (10 components), neighbors k=2 : 20, on real world datasets. p=2 Euclidean, p=1 Manhattan, $p \to \infty$ Chebyshev, C Canberra.

were designed to use by default the Euclidean metric, and in some cases as the only available metric.

Our results provided insights to help in the selection process of a distance metric when interacting with factors such as algorithm, parametrization, data size and dimensionality. Moreover, we attempted to provide a mechanism for selecting a distance metric, not by merely blindly selecting the metric with possibly the highest AUC or the lowest processing time, but instead, by guiding in the intricate combination of algorithm - parametrization - data. Our study revealed that beyond basic and common knowledge about how the size and dimensionality of data or the number of algorithms in an ensemble influence the detection rate and processing time, there are factors, such as the selection of a distance metric, that further influence these elements. It is our aim that such insights will be beneficial in the selection and parametrization of an outlier detector when working on real-world domains and that they will also contribute to the development of new algorithms for outlier detection.

4.7 Conclusions and future work

In this study, we examined how either a single detector or an ensemble of outlier detectors can be affected by the selection of a specific distance metric, considering factors like the data size, dimensionality, parameter settings, ensemble components, etc. Our study provides a solid foundation for further research covering a broader set of scenarios, and more importantly, it provides critical insights to be used in the selection and parametrization of a detector or an ensemble of detectors for unsupervised outlier detection.

Only two approaches, LOF and feature bagging, were examined in this study, but we expect our results to be generalizable to similar algorithms, based either on distances or densities, and that they will serve as a reference for the selection of a distance metric for such approaches. Interestingly, Euclidean distance has generally been used as the default metric in most of the algorithms whose main mechanism depends on the computation of distances. Our results suggest that although it is in general relatively straightforward to get positive results with an unweighted

Minkowski metric, in real-world scenarios a weighted version of Minkowski can offer a similar detection rate, and in some scenarios, rates that are even higher than that of its unweighted version.

Although our experiments attempted to cover different data scenarios and parameter settings, further evaluation is needed, considering a wider set of algorithms, datasets, parameters and more importantly a larger set of distance metrics, in future work we will address these issues.

CHAPTER 5

GENERAL DISCUSSION

This thesis has addressed the general problem of unsupervised outlier detection. There are three main challenges encountered in outlier detection, namely, the heterogeneity of outliers, the hidden outlier behavior of interesting observations and the parameterization of an outlier detector; **Chapter 1** presents a review of the literature in outlier detection and specifically the limitations of current approaches. The introduction section established three research objectives to address the previous problems. Subsequently, **Chapters 2**, **3** and **4** proposed novel approaches to address our research objectives. First, a novel approach for the detection of heterogeneous types of outliers was proposed (**Chapter 2**). Second, an ensemble mechanism to detect outliers hidden in lower dimensional subspaces was developed (**Chapter 3**). Finally, a guide for the selection of a distance metric based on specific parameter settings was established (**Chapter 4**). Although the contributions in this thesis address independent problems, they are also complementary. In the following sections, we discuss them with a global perspective, focusing on their complementarity, possible uses, advantages and disadvantages.

5.1 Detection of outliers using heterogeneous types of detectors

Most of the outlier detection algorithms are oriented towards the detection of a specific type of outlier. Such behavior is not explicit, instead it is implicit in the mechanisms used by the algorithm (e.g. extreme value detection methods and clustering based methods are able to detect only outliers in the tails of a distribution, or as points that are far away from the main clusters in the data, respectively). Moreover, the unsupervised and unbalanced nature of outlier detection represents a challenge when merging results from different algorithms, without labeled data it is impossible to select the best performing algorithms based on external validation measures (e.g. precision, recall or accuracy), this has led to a lack of approaches oriented towards this scenario.

In **Chapter 2** we proposed two unsupervised ensemble mechanisms (EDCV and EDVV) to combine scores from different types of detectors. Both approaches are able to operate in a fully unsupervised setting, assigning distinct weights to each algorithm in the ensemble depending on two internal validation measures. The difference between them is the mechanism used to build the vectors of weights, the former construct the vector of weights by computing the correlation between the results of the different components in the ensemble, the latter builds a similar vector of weights by computing the mean absolute deviation between each pair of vectors. Both approaches improve the detection rate of a single classifier and even that of similar ensemble approach; however, their improvements in detection rate are also followed by an increase in processing time, linearly dependent on the number of ensemble components, also, the outliers hidden in lower dimensional spaces are neglected. These limitations were addressed by our subsequent approach.

EDCV and EDVV make use of a voting mechanism in order to improve the differentiation between outliers and outliers. It is important to note that a voting mechanism could be biased due to the equally importance that both approaches assign to each detector when considering the number of votes. Despite that this factor is addressed by the weighted mechanisms used by both approaches, a specific dataset where most of the base algorithms exhibit a extremely poor performance can induce a small deterioration in detection rate. In this case our two proposed approaches can be used with or without this voting system, thus its use remains application dependent.

5.2 Detection of outliers in lower-dimensional spaces

In **Chapter 3** we addressed two issues not considered by our previous approaches: the outliers hidden in lower-dimensional projections and the processing time of an ensemble of detectors. An outlier detector generally searches for outliers in full dimensional space; however, interesting outliers are usually located in specific subsets of dimensions, then by using a full-dimensional detector their outlier behavior remains masked. Thus, in **Chapter 3**, we developed an ensemble approach to search for those hidden outliers while avoiding the high-processing

times usually found in an ensemble of detectors. The proposed approach, feature bagged subspaces for outlier detection (FBSO) is based in two internal mechanisms. First, it uses random sets of dimensions of variable size between $d/2$ and d-1, being d the dimensionality of the data. Second, it randomly samples observations without replacements from the data. These two random mechanisms not only provided and improvement in detection rate (increasing the individual variance, but improving detection rate), but also reduce the processing time of the ensemble.

It is worth to mention that similarly to our proposed combination mechanism in **Chapter 2**, FBSO overlooked the effect of the parameterization of an algorithm in detection rate and processing time. In the next section we aimed to provide a deeper understanding of the multiple interaction between data, parameterization and algorithm.

5.3 Interaction of algorithm's parameters and data

The approaches proposed in **Chapters 2** and **3** addressed independent, but complementary problems, the weighted combination of scores from distinct outlier detectors and the propensity of outliers to hide in lower-dimensional projections of the data, respectively. These approaches provided an improved detection rate, inducing diversity either with the use of different types of detectors or with variations in the search space. Moreover, the approach in **Chapter 3** also reduced the expected processing time of an ensemble of detectors; however, they did not considered how all the specific combinations of data, algorithms and parameters could affect detection rate and processing time. While most of the current unsupervised ensemble approaches for outlier detection considered in their evaluation the use of different sizes and dimensionalities of data, quantity of ensemble components and in some cases even the number of nearest neighbors, the effect that distinct distance metrics have in the interaction data, algorithm and parameterization remained hidden. Moreover, the parameterization of an outlier detection algorithm is a pervasive and overlooked problem in outlier detection. Thus, in **Chapter 4** we explored the interaction of different distance metrics with distinct dimensionalities, data sizes, algorithms, etc. This study revealed some of the strengths and weaknesses of distance met-

rics, thus providing interesting insights for the selection of a specific distance metric in the unsupervised outlier detection scenario.

CONCLUSION AND RECOMMENDATIONS

In many domains, important events are not represented as the common scenario, but as deviations from the rule. Outlier detection algorithms have been designed to detect these deviant, outnumbered and hidden events. Most of current approaches for outlier detection are based on strong assumptions about a specific type of outlier or were designed to find outliers in full dimensionality. However, a single dataset can contain different types of outliers which are not easily identifiable by a single outlier detector; moreover, differently from trivial outliers which are usually located in full dimensionality, interesting observations exhibit their outlier behavior only in a specific subset of dimensions. The unsupervised and unbalanced nature of outlier detection represents a challenge for the detection of this heterogeneous and hidden outliers, as well as for the identification of the most appropriate algorithm's parameters for a specific dataset.

In this thesis, we have addressed the unsupervised, unbalanced, diverse and hidden nature of outliers. Two approaches for an unsupervised weighted combination of different types of detectors were proposed. Moreover, an ensemble algorithm to detect outliers in lower-dimensional subspaces was developed. Finally, a guide in the parameterization of an outlier detector was established.

In **Chapter 1**, a review of current approaches for outlier detection is described. A relatively vast number of approaches have been proposed for outlier detection. However, these approaches are generally based on strong assumptions about what constitutes an outlier, which results in a lack of approaches oriented towards the identification of different types of deviant observations. This section also highlights the lack of computational inexpensive approaches for the detection of outliers hidden in lower-dimensional projections of the data. Moreover, current studies in outlier detection usually consider only a limited set of algorithms' parameters and data

characteristics in their experiments, with an absence of a more complete set experiments which could provide deeper insights in the intricate interaction detector - parametrization - data.

In **Chapter 2**, two mechanisms for the combination of scores from different types of detectors are described. Both combination functions, EDCV and EDVV, are based on unsupervised procedures to assign weights depending on the ability of a specific algorithm over the dataset at hand. Outlier detection algorithms are based on strong and distinct assumptions about the characteristics that define an outlier. EDCV and EDVV leverage this variety of perspectives to produce a diverse and potentially more robust ensemble.

In **Chapter 3**, an unsupervised ensemble algorithm, FBSO, for the detection of outliers hidden in lower-dimensional spaces is proposed. FBSO introduced the combined use of two mechanisms to induce diversity in the ensemble, thus lowering variance and improving detection rate. The approach is able to detect observations whose outlier behavior is revealed only on specific, but unknown subsets of dimensions, by using an iterative random selection process. The algorithm further increases diversity by using random samples of data in which the scores of each observation are computed. The use of these sampling procedures in combination with a density-based method tend to produce diverse and potentially complementary outputs, which results in a more robust classifier. Thus, FBSO uses this reduced dataset, both in dimensions and observations, to compute density estimates which are based in different sets of neighbors, producing a more robust classifier.

In **Chapter 4**, the interaction between data, outlier detector and parameters settings has been investigated and applied to synthetic and real-world datasets. The set of experiments considered factors like data size, dimensionality, distance metrics, parameter settings and ensemble components. A comparison between different types of distance metrics showed that despite the prevalent use of the Euclidean distance as the default metric in most of the distance-based outlier detection algorithms, in real-world scenarios a weighted version of a Minkowski met-

ric performs similarly or even better than a unweighted metric like the Euclidean. The results in this study cover a gap in the outlier detection literature by providing a mechanism for the selection and parameterization of an unsupervised outlier detector.

Future work

Knowledge about the individual characteristics that define a specific observation as an outlier, which is generally known as intentional knowledge, remains as an open question. Outlier detection is all about providing more insights with less or almost no information about the underlying data, current approaches are only capable of unveiling the identity of the outliers to the final user. The lack of approaches oriented towards intentional knowledge is mainly due to inherent characteristics of the outliers that the user aims to detect, namely unsupervised, unbalanced and hidden outliers. Our approach described in **Chapter 3** is capable of locating outliers hidden in lower-dimensional projections of the data with a relatively low execution time, we expect to develop an improved version of this ensemble approach capable of identifying the specific attributes in which the different outliers exhibit their abnormal behavior, these results would provide a complete view of the potential outliers found in a dataset, allowing the user to decide about the actions to be taken before attempting a deeper analysis. Moreover, we will apply the approaches developed in this thesis to the education domain, with the aim to detect potentially future outlier behavior in students. Finally, based in the literature described in **Chapter 1** and on an extended version of the set of experiments in **Chapter 4** we will seek to develop a survey that serves as a final model for the selection, parameterization and interpretation of an outlier detector depending on the dataset under study.

BIBLIOGRAPHY

Aggarwal, C. C. (2005). On abnormality detection in spuriously populated data streams. *Proceedings of the 2005 siam international conference on data mining*, pp. 80–91.

Aggarwal, C. C. (2013a). Outlier ensembles: position paper. *Sigkdd explor. newsl.*, 14(2), 49-58. doi: 10.1145/2481244.2481252.

Aggarwal, C. C. (2013b). *Outlier analysis*. Springer Science & Business Media.

Aggarwal, C. C. (2015). Outlier analysis. *Data mining*, pp. 237–263.

Aggarwal, C. C. & Sathe, S. (2015). Theoretical foundations and algorithms for outlier ensembles. *Acm sigkdd explorations newsletter*, 17(1), 24-47.

Aggarwal, C. C. & Yu, P. S. (2001). Outlier detection for high dimensional data. *Sigmod rec.*, 30(2), 37-46. doi: 10.1145/376284.375668.

Aggarwal, C., Hinneburg, A. & Keim, D. (2001). On the surprising behavior of distance metrics in high dimensional space. In *Database Theory ICDT 2001* (vol. 1973, ch. 27, pp. 420-434). Springer Berlin Heidelberg.

Agrawal, R., Gehrke, J., Gunopulos, D. & Raghavan, P. (1998). *Automatic subspace clustering of high dimensional data for data mining applications*. ACM.

Angiulli, F. & Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. *European conference on principles of data mining and knowledge discovery*, pp. 15–27.

Angiulli, F. & Pizzuti, C. (2005). Outlier mining in large high-dimensional data sets. *Ieee transactions on knowledge and data engineering*, 17(2), 203–215.

Angiulli, F., Fassetti, F. & Palopoli, L. (2009). Detecting outlying properties of exceptional objects. *Acm transactions on database systems (tods)*, 34(1), 7.

Arning, A., Agrawal, R. & Raghavan, P. (1996). A linear method for deviation detection in large databases. *Kdd*, pp. 164–169.

Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [Dataset].

Barnett, V. (1976). The ordering of multivariate data. *Journal of the royal statistical society. series a (general)*, 318–355.

Barnett, V. & Lewis, T. (1994). *Outliers in statistical data*. Wiley New York.

Beckman, R. J. & Cook, R. D. (1983). Outliers. *Technometrics*, 25(2), 119–149.

Beyer, K., Goldstein, J., Ramakrishnan, R. & Shaft, U. (1999). When is "nearest neighbor" meaningful? *International conference on database theory*, pp. 217–235.

Bickel, S. & Scheffer, T. (2004). Multi-view clustering. *Proceedings of the fourth ieee international conference on data mining*, pp. 19–26.

Birant, D. & Kut, A. (2007). St-dbscan: An algorithm for clustering spatial–temporal data. *Data & knowledge engineering*, 60(1), 208–221.

Blum, A. & Rivest, R. L. (1989). Training a 3-node neural network is np-complete. *Advances in neural information processing systems*, pp. 494–501.

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7), 1145–1159.

Branch, J. W., Giannella, C., Szymanski, B., Wolff, R. & Kargupta, H. (2013). In-network outlier detection in wireless sensor networks. *Knowledge and information systems*, 34(1), 23–54.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2), 123–140.

Breunig, M., Kriegel, H.-P., Ng, R. & Sander, J. (1999). Optics-of: Identifying local outliers. *Principles of data mining and knowledge discovery*, 262–270.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. & Sander, r. (2000). Lof: identifying density-based local outliers. *Sigmod rec.*, 29(2), 93-104. doi: 10.1145/335191.335388.

Brown, G. (2011). Ensemble learning. In *Encyclopedia of Machine Learning* (pp. 312–320). Springer.

Brown, G., Wyatt, J., Harris, R. & Yao, X. (2005). Diversity creation methods: a survey and categorisation. *Information fusion*, 6(1), 5–20.

Campos, G. O., Zimek, A., Sander, J., Campello, R. J., Micenková, B., Schubert, E., Assent, I. & Houle, M. E. (2015). On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data mining and knowledge discovery*, 1–37.

Canuto, A. M., Abreu, M. C., de Melo Oliveira, L., Xavier, J. C. & Santos, A. d. M. (2007). Investigating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensembles. *Pattern recognition letters*, 28(4), 472–486.

Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *International journal of mathematical models and methods in applied sciences*.

Chandola, V., Banerjee, A. & Kumar, V. (2009). Anomaly detection: A survey. *Acm comput. surv.*, 41(3), 1-58. doi: 10.1145/1541880.1541882.

Chandra, A., Chen, H. & Yao, X. (2006). Trade-off between diversity and accuracy in ensemble generation. In *Multi-objective machine learning* (pp. 429–464). Springer.

Chen, Z., Tang, J. & Fu, A. W.-C. (2003). Modeling and efficient mining of intentional knowledge of outliers. *Database engineering and applications symposium, 2003. proceedings. seventh international*, pp. 44–53.

Craswell, N. (2009). R precision. In *Encyclopedia of Database Systems* (pp. 2453–2453). Springer.

Cunningham, P. & Delany, S. J. (2007). k-nearest neighbour classifiers. *Multiple classifier systems*, 34, 1–17.

Dang, X. H., Assent, I., Ng, R. T., Zimek, A. & Schubert, E. (2014). Discriminative features for identifying and interpreting outliers. *Data engineering (icde), 2014 ieee 30th international conference on*, pp. 88–99.

Das, S., Wong, W.-K., Dietterich, T., Fern, A. & Emmott, A. (2016). Incorporating expert feedback into active anomaly discovery. *Data mining (icdm), 2016 ieee 16th international conference on*, pp. 853–858.

Dasgupta, D. & Nino, F. (2000). A comparison of negative and positive selection algorithms in novel pattern detection. *Systems, man, and cybernetics, 2000 ieee international conference on*, 1, 125–130.

Desforges, M., Jacob, P. & Cooper, J. (1998). Applications of probability density estimation to the detection of abnormal conditions in engineering. *Proceedings of the institution of mechanical engineers, part c: Journal of mechanical engineering science*, 212(8), 687–703.

Ditterrich, T. (1997). Machine learning research: four current direction. *Artificial intelligence magzine*, 4, 97–136.

Domeniconi, C., Papadopoulos, D., Gunopulos, D. & Ma, S. (2004). Subspace clustering of high dimensional data. *Proceedings of the 2004 siam international conference on data mining*, pp. 517–521.

Domingos, P. (2016). *Master algorithm*. Penguin Books.

Edgeworth, F. (1887). Xli. on discordant observations. *The london, edinburgh, and dublin philosophical magazine and journal of science*, 23(143), 364–375.

Emmott, A. F., Das, S., Dietterich, T., Fern, A. & Wong, W.-K. (2013). *Systematic construction of anomaly detection benchmarks from real data*. Conference Paper presented in Proceedings of the acm sigkdd workshop on outlier detection and description (pp. 16-21). doi: 10.1145/2500853.2500858.

Eskin, E. (2000). Anomaly detection over noisy data using learned probability distributions. *Proceedings of the seventeenth international conference on machine learning*, pp. 255–262.

Eskin, E., Arnold, A., Prerau, M., Portnoy, L. & Stolfo, S. (2002). A geometric framework for unsupervised anomaly detection. In *Applications of data mining in computer security* (pp. 77–101). Springer.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X. et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*, 96(34), 226–231.

Fawcett, T. (2004). Roc graphs: Notes and practical considerations for researchers. *Machine learning*, 31, 1-38.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8), 861–874.

Filzmoser, P., Maronna, R. & Werner, M. (2008). Outlier identification in high dimensions. *Computational statistics & data analysis*, 52(3), 1694–1711.

Freund, Y. & Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. *European conference on computational learning theory*, pp. 23–37.

Gao, J. & Tan, P.-N. (2006). Converting output scores from outlier detection algorithms into probability estimates. *Data mining, 2006. icdm'06. sixth international conference on*, pp. 212–221.

Ghosh, J. & Acharya, A. (2011). Cluster ensembles. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 1(4), 305-315.

Gionis, A., Mannila, H. & Tsaparas, P. (2007). Clustering aggregation. *Acm transactions on knowledge discovery from data (tkdd)*, 1(1), 4.

Grubbs, F. E. (1969). Procedures for detecting outlying observations in samples. *Technometrics*, 11(1), 1-21. doi: 10.1080/00401706.1969.10490657.

Hartigan, J. A. & Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1), 100–108.

Hawkins, D. M. (1980). *Identification of outliers*. Springer.

He, Z., Deng, S. & Xu, X. (2005). A unified subspace outlier ensemble framework for outlier detection. *Advances in web-age information management*, 632–637.

Helman, P. & Bhangoo, J. (1997). A statistically based system for prioritizing information exploration under uncertainty. *Ieee transactions on systems, man, and cybernetics-part a: Systems and humans*, 27(4), 449–466.

Hinneburg, A., Aggarwal, C. C. & Keim, D. A. (2000). *What is the nearest neighbor in high dimensional spaces*. Book presented in Proceedings of the26th vldb conference, cairo, egypt, 2000.

Hodge, V. J. & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2), 85–126.

Hsu, K.-W. & Srivastava, J. (2009). Diversity in combinations of heterogeneous classifiers. *Advances in knowledge discovery and data mining*, 923–932.

Huang, B. & Yang, P. (2011). Finding key knowledge attribute subspace of outliers in high-dimensional dataset. *Expert systems with applications*, 38(8), 10147–10152.

Huang, J. & Ling, C. X. (2005). Using auc and accuracy in evaluating learning algorithms. *Ieee transactions on knowledge and data engineering*, 17(3), 299–310.

Inatani, S. & Suzuki, E. (2002). Data squashing for speeding up boosting-based outlier detection. *International symposium on methodologies for intelligent systems*, pp. 601–611.

Irani, J., Pise, N. & Phatak, M. (2016). Clustering techniques and the similarity measures used in clustering: A survey. *International journal of computer applications*, 134(7), 9–14.

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2015). *An introduction to statistical learning: with applications in r*. Springer New York. doi: 10.1007/978-1-4614-7138-7.

Javitz, H. S. & Valdes, A. (1991). The sri ides statistical anomaly detector. *Research in security and privacy, 1991. proceedings., 1991 ieee computer society symposium on*, pp. 316–326.

Jin, W., Tung, A. K. & Han, J. (2001). Mining top-n local outliers in large databases. *Proceedings of the seventh acm sigkdd international conference on knowledge discovery and data mining*, pp. 293–298.

Jin, W., Tung, A. K., Han, J. & Wang, W. (2006). Ranking outliers using symmetric neighborhood relationship. *Pakdd*, 6, 577–593.

Jing, G. & Pang-Ning, T. (2006). *Converting output scores from outlier detection algorithms into probability estimates*. Conference Proceedings presented in Data mining, 2006. icdm '06. sixth international conference on (pp. 212-221). doi: 10.1109/icdm.2006.43.

Johnson, T., Kwok, I. & Ng, R. T. (1998). Fast computation of 2-dimensional depth contours. *Kdd*, pp. 224–228.

Jolliffe, I. (2002). *Principal component analysis*. Wiley Online Library.

Joshi, M. V. & Kumar, V. (2004). Credos: Classification using ripple down structure (a case for rare classes). *Sdm*, pp. 321–332.

Kamber, M., Han, J. & Pei, J. (2012). *Data mining: Concepts and techniques*. Elsevier.

Kelleher John D.,Brian Mac Namee, A. D. (2015). *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT Press.

Keller, F., Muller, E. & Bohm, K. (2012). *Hics: High contrast subspaces for density-based outlier ranking*. Conference Proceedings presented in Data engineering (icde), 2012 ieee 28th international conference on (pp. 1037-1048). doi: 10.1109/ICDE.2012.88.

Khoshgoftaar, T. M., Nath, S. V., Zhong, S. & Seliya, N. (2005). Intrusion detection in wireless networks using clustering techniques with expert analysis. *Machine learning and applications, 2005. proceedings. fourth international conference on*, pp. 6–pp.

Knorr, E. M. & Ng, R. T. (1997). A unified notion of outliers: Properties and computation. *Kdd*, pp. 219–222.

Knorr, E. M. & Ng, R. T. (1999). *Finding intensional knowledge of distance-based outliers*. Conference Proceedings presented in Vldb (pp. 211-222).

Knorr, E. M., Ng, R. T. & Tucakov, V. (2000). Distance-based outliers: algorithms and applications. *The vldb journal—the international journal on very large data bases*, 8(3-4), 237–253.

Knox, E. M. & Ng, R. T. (1998). Algorithms for mining distancebased outliers in large datasets. *Proceedings of the international conference on very large data bases*, pp. 392–403.

Korn, F., Labrinidis, A., Labrinidis, R., Kotidis, Y., Faloutsos, C., Perkovic, D. & Kaplunovich, A. (1997). Quantifiable data mining using principal component analysis. *Vldb journal: Very large data bases*.

Kriegel, H.-P., Zimek, A. et al. (2008). Angle-based outlier detection in high-dimensional data. *Proceedings of the 14th acm sigkdd international conference on knowledge discovery and data mining*, pp. 444–452.

Kriegel, H.-P., Kröger, P., Schubert, E. & Zimek, A. (2009a). Loop: local outlier probabilities. *Proceedings of the 18th acm conference on information and knowledge management*, pp. 1649–1652.

Kriegel, H.-P., Kröger, P. & Zimek, A. (2009b). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *Acm transactions on knowledge discovery from data (tkdd)*, 3(1), 1.

Kriegel, H.-P., Kröger, P., Schubert, E. & Zimek, A. (2011). Interpreting and unifying outlier scores. In *Proceedings of the 2011 SIAM International Conference on Data Mining* (pp. 13–24).

Kuncheva, L. I. (2003). That elusive diversity in classifier ensembles. *Iberian conference on pattern recognition and image analysis*, pp. 1126–1138.

Laurikkala, J., Juhola, M., Kentala, E., Lavrac, N., Miksch, S. & Kavsek, B. (2000). Informal identification of outliers in medical data. *Fifth international workshop on intelligent data analysis in medicine and pharmacology*, pp. 20–24.

Lazarevic, A. & Kumar, V. (2005). *Feature bagging for outlier detection*. Conference Proceedings presented in Conference on knowledge discovery in data: Proceeding of the eleventh acm sigkdd international conference on knowledge discovery in data mining (pp. 157-166).

Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A. & Srivastava, J. (2003). *A comparative study of anomaly detection schemes in network intrusion detection*. Conference Proceedings presented in Proceedings of the third siam international conference on data mining (pp. 25-36).

Leckie, C. (2016). Smart sampling: A novel unsupervised boosting approach for outlier detection. *Ai 2016: Advances in artificial intelligence: 29th australasian joint conference, hobart, tas, australia, december 5-8, 2016, proceedings*, 9992, 469.

Lichman, M. (2013). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. Consulted at http://archive.ics.uci.edu/ml.

Lin, X. & Chen, X.-w. (2010). Mr. knn: soft relevance for multi-label classification. *Proceedings of the 19th acm international conference on information and knowledge management*, pp. 349–358.

Liu, F. T., Ting, K. M. & Zhou, Z.-H. (2012). Isolation-based anomaly detection. *Acm trans. knowl. discov. data*, 6(1), 1-39. doi: 10.1145/2133360.2133363.

Marques, H. O., Campello, R. J., Zimek, A. & Sander, J. (2015). On the internal evaluation of unsupervised outlier detection. *Proceedings of the 27th international conference on scientific and statistical database management*, pp. 7.

Muller, E., Gunnemann, S., Fa, x, rber, I. & Seidl, T. (2012a). *Discovering multiple clustering solutions: Grouping objects in different views of the data*. Conference Proceedings presented in Data engineering (icde), 2012 ieee 28th international conference on (pp. 1207-1210). doi: 10.1109/ICDE.2012.142.

Müller, E., Schiffer, M. & Seidl, T. (2011). Statistical selection of relevant subspace projections for outlier ranking. *Data engineering (icde), 2011 ieee 27th international conference on*, pp. 434–445.

Muller, E., Assent, I., Iglesias, P., Mulle, Y. & Bohm, K. (2012b). *Outlier ranking via subspace analysis in multiple views of the data*. Conference Proceedings presented in Data mining (icdm), 2012 ieee 12th international conference on (pp. 529–538).

Ng, R. & Han, J. (1994). Efficient and effective clustering algorithms for spatial data mining. *Proceedings of the 20th international conference on very large data bases*, pp. 144–155.

Nguyen, H., Ang, H. & Gopalkrishnan, V. (2010). Mining outliers with ensemble of heterogeneous detectors on random subspaces. In *International Conference on Database*

Systems for Advanced Applications (vol. 5981, pp. 368-383). Springer Berlin / Heidelberg. doi: 10.1007/978-3-642-12026-8_29.

Noto, K., Brodley, C. & Slonim, D. (2010). Anomaly detection using an ensemble of feature models. *2010 ieee international conference on data mining*, pp. 953–958.

Opitz, D. & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11, 169–198.

Oza, N. C. & Tumer, K. (2008). Classifier ensembles: Select real-world applications. *Information fusion*, 9(1), 4–20.

Palpanas, T., Papadopoulos, D., Kalogeraki, V. & Gunopulos, D. (2003). Distributed deviation detection in sensor networks. *Acm sigmod record*, 32(4), 77–82.

Papadimitriou, S., Kitagawa, H., Gibbons, P. B. & Faloutsos, C. (2003). Loci: Fast outlier detection using the local correlation integral. *Data engineering, 2003. proceedings. 19th international conference on*, pp. 315–326.

Parsons, L., Haque, E. & Liu, H. (2004). Subspace clustering for high dimensional data: a review. *Acm sigkdd explorations newsletter*, 6(1), 90–105.

Pasillas-Díaz, J. R. & Ratté, S. (2016a). Bagged subspaces for unsupervised outlier detection. *Computational intelligence*.

Pasillas-Díaz, J. R. & Ratté, S. (2016b). An unsupervised approach for combining scores of outlier detection techniques, based on similarity measures. *Electronic notes in theoretical computer science*, 329, 61–77.

Patcha, A. & Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer networks*, 51(12), 3448–3470.

Piepel, G. F. (1989). Robust regression and outlier detection. *Technometrics*, 31(2), 260–261.

Ramaswamy, S., Rastogi, R. & Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. *Acm sigmod record*, 29, 427–438.

Rayana, S. & Akoglu, L. (2016). Less is more: Building selective anomaly ensembles. *Acm transactions on knowledge discovery from data (tkdd)*, 10(4), 42.

Rokach, L. (2009). Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Computational statistics & data analysis*, 53(12), 4046–4072.

Rousseeuw, P. J. & Hubert, M. (2011). Robust statistics for outlier detection. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 1(1), 73–79.

Rousseeuw, P. J. & Leroy, A. M. (2005). *Robust regression and outlier detection*. John wiley & sons.

Ruts, I. & Rousseeuw, P. J. (1996). Computing depth contours of bivariate point clouds. *Computational statistics & data analysis*, 23(1), 153–168.

Schubert, E., Wojdanowski, R., Zimek, A. & Kriegel, H.-P. (2012). *On evaluation of outlier rankings and outlier scores*. Conference Proceedings presented in Proceedings of the 12th siam international conference on data mining (sdm), anaheim, ca, 2012 (pp. 1047-1058).

Schubert, E., Zimek, A. & Kriegel, H.-P. (2014a). Generalized outlier detection with flexible kernel density estimates. In *Proceedings of the 2014 SIAM International Conference on Data Mining* (vol. 14, pp. 542–550).

Schubert, E., Zimek, A. & Kriegel, H.-P. (2014b). Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. *Data mining and knowledge discovery*, 28(1), 190–237.

Soares, R. G., Santana, A., Canuto, A. M. & de Souto, M. C. P. (2006). Using accuracy and diversity to select classifiers to build ensembles. *Neural networks, 2006. ijcnn'06. international joint conference on*, pp. 1310–1316.

Tan, K. M. & Maxion, R. A. (2005). The effects of algorithmic diversity on anomaly detector performance. *2005 international conference on dependable systems and networks (dsn'05)*, pp. 216–225.

Tang, J., Chen, Z., Fu, A. & Cheung, D. (2002). Enhancing effectiveness of outlier detections for low density patterns. *Advances in knowledge discovery and data mining*, 535–548.

Torgo, L. (2007). Resource-bounded fraud detection. *Portuguese conference on artificial intelligence*, pp. 449–460.

Torgo, L. (2010). *Data mining with r: learning with case studies*. Chapman & Hall/CRC.

Tukey, J. W. (1977). Exploratory data analysis. *Addison-wesley series in behavioral science: Quantitative methods, reading, mass.: Addison-wesley, 1977*, 1.

Tumer, K. & Ghosh, J. (1996). Analysis of decision boundaries in linearly combined neural classifiers. *Pattern recognition*, 29(2), 341–348.

Wang, W., Zhang, J. & Wang, H. (2005). Grid-odf: detecting outliers effectively and efficiently in large multi-dimensional databases. *Computational intelligence and security*, 765–770.

Windeatt, T. (2005). Diversity measures for multiple classifier system analysis and design. *Information fusion*, 6(1), 21–36.

Xuan Hong, D., Assent, I., Ng, R. T., Zimek, A. & Schubert, E. (2014). *Discriminative features for identifying and interpreting outliers*. Conference Proceedings presented in Data engineering icde, 2014 ieee 30th international conference on (pp. 88-99). doi: 10.1109/ICDE.2014.6816642.

Yang, P. & Zhu, Q. (2011). Finding key attribute subset in dataset for outlier detection. *Knowledge-based systems*, 24(2), 269–274.

Zhang, J. (2013). Advancements of outlier detection: A survey. *Eai endorsed trans. scalable information systems*, 1(1), e2.

Zhang, T., Ramakrishnan, R. & Livny, M. (1996). Birch: An efficient data clustering method for very large databases. *Sigmod rec.*, 25(2), 103–114. doi: 10.1145/235968.233324.

Zimek, A., Schubert, E. & Kriegel, H.-P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical analysis and data mining*, 5(5), 363–387.

Zimek, A., Gaudet, M., Campello, R. J. & Sander, J. (2013). *Subsampling for efficient and effective unsupervised outlier detection ensembles*. Conference Proceedings presented in Proceedings of the 19th acm sigkdd international conference on knowledge discovery and data mining (pp. 428–436).

Zimek, A., Campello, R. J., & Sander, r. (2014). Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *Sigkdd explor. newsl.*, 15(1), 11-22. doi: 10.1145/2594473.2594476.