Interprétation d'espaces de représentation: caractérisation de métriques quantitatives et recommandations

par

Jonathan BOILARD

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN GÉNIE ÉLECTRIQUE
M. Sc. A.

MONTRÉAL, LE 30 MAI 2022

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Ghyslain Gagnon, directeur de mémoire
Département de génie électrique à l'École de technologie supérieure

Mme Rita Noumeir, présidente du jury
Département de génie électrique à l'École de technologie supérieure

M. Christian Desrosiers, professeur
Département de génie électrique à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 29 AVRIL 2022

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

Interprétation d'espaces de représentation: Caractérisation de métriques quantitatives et recommandations

Jonathan BOILARD

RÉSUMÉ

Plusieurs ouvrages proposent des modèles d'apprentissage non supervisés permettant de découvrir une variété de représentations génériques d'un ensemble de données. Cependant, l'optimisation de la fonction objective de ces modèles n'assure pas l'obtention de représentations démêlées, soit explicitement utiles sur des tâches connexes ultérieures. Afin de comparer efficacement différentes représentations obtenues, une méthode pour mesurer quantitativement le démêlage est nécessaire. Diverses métriques appropriées à cette problématique ont été proposées. Cependant, il est observé qu'elles sont souvent incohérentes lorsqu'elles sont comparées l'une à l'autre ou comparées à l'évaluation subjective du praticien. Comparer les métriques s'avère difficile dans un contexte typique d'apprentissage de représentations, puisque la nature générique des représentations obtenues empêche de connaître avec certitude la réelle qualité des propriétés mesurées. Afin de rendre les métriques fiables, il est important de démystifier ces incohérences.

Afin de remédier à ce problème, cet ouvrage propose de caractériser les métriques sur des représentations dont les propriétés représentatives sont connues. Une taxonomie est d'abord mise en place permettant d'identifier les similarités entre les métriques. Cette taxonomie n'est cependant pas suffisante pour comprendre le désaccord entre les métriques. Des propriétés de métriques désirables sont définies, et les scénarios proposés servent à procéder à la caractérisation en fonction de ces propriétés.

Dans ce document, il est découvert que plusieurs métriques ont de la difficulté à correctement mesurer des propriétés dont elles devraient être capables de fournir des mesures. Nous identifions DCI comme la plus robuste à l'identification du démêlage selon la qualité explicite, la modularité et la compacité d'une représentation. Dans nos scénarios représentatifs expérimentaux, DCI évite plusieurs instabilités de causes diverses. DCI peut donc être utilisée sans crainte relativement à sa compatibilité à l'ensemble de données et la représentation dans laquelle le praticien désire y mesurer le démêlage. Finalement, nous discutons des différences clés entre les ensembles expérimentaux et réels de données ainsi que différentes considérations pratiques afin d'identifier de futures pistes d'amélioration.

Mots clés : Apprentissage machine, Apprentissage de représentations, démêlage, métriques

Disentangled Latent spaces: Characterisation of Supervised metrics and recommendations

Jonathan BOILARD

ABSTRACT

Several studies offer unsupervised learning models that allow one to discover a variety of generic dataset representations. However, optimizing the objective function of these models does not ensure that disentangled representations are obtained that are explicitly useful on subsequent related tasks. In order to effectively compare different representations obtained, a method for quantitatively measuring disentanglement is needed. Various metrics addressing this problem have been proposed. However, it is observed that they are often inconsistent when compared to each other or compared to a practitioner's subjective assessment. Comparing metrics is difficult in a typical representation learning context, since the generic nature of the representations obtained prevents knowing with certainty the real quality of the measured properties. In order to make metrics reliable, it is important to demystify these inconsistencies.

In order to remedy this problem, this work proposes to characterize the metrics on representations whose representative properties are known. First, a metric taxonomy is put in place to help identify the similarities between them. This taxonomy, however, is not sufficient to understand the disagreement between metrics. Desirable metric properties are defined, and the proposed scenarios are used to characterize against these properties.

In this document, it is discovered that several metrics have difficulty in correctly measuring properties of which they should be able to provide measurements. We identify DCI as the most robust in identifying disentangling according to the explicit quality, modularity and compactness of a representation. In our representative experimental scenarios, DCI avoids several instabilities of various causes. DCI can therefore be used without fear of its compatibility with the data set and the representation in which the practitioner wishes to measure disentanglement therein. Finally, we further discuss key differences between experimental and actual data sets as well as various practical considerations & identify further possibilities for improvement in future studies.

Keywords : Machine learning, Representation learning, Disentanglement, Metric

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

LISTE DES SYMBOLES ET UNITÉS DE MESURE

ENSEMBLES	
$D^{(N)}$	Agglomération $\{V, X, Z\}^{(N)}$ de l'ensemble des facteurs $V^{(N)}$, des observations $X^{(N)}$ et des codes $Z^{(N)}$.
$V^{(N)}$	Ensemble de facteurs $\boldsymbol{v} \in V^{(N)}$
$Z^{(N)}$	Ensemble de codes $\boldsymbol{z} \in Z^{(N)}$
$X^{(N)}$	Ensemble de codes $\boldsymbol{x} \in X^{(N)}$
$\{\mathcal{V}, \mathcal{Z}\}^{(\mathcal{N})}$	Sous-ensembles interventionnels échantillonnés de $\{V, Z\}^{(N)}$, tel que $\{\mathcal{V}, \mathcal{Z}\}^{(\mathcal{N})} \subset \{V, Z\}^{(N)}$
VECTEURS ET ELÉMENTS DE VECTEURS	
$\boldsymbol{v}$	Contient M facteurs $\boldsymbol{v} = \{v_1, v_2, \dots, v_M\}$
$\boldsymbol{z}$	Code à d dimensions $\boldsymbol{z} = \{z_1, z_2, \dots, z_d\}$
v_i	Valeur de facteurs à l'index i, soit $v_i \in \{v_1, v_2, \dots, v_M\}$
z_j	Une dimension des codes $\boldsymbol{z} = \{z_1, z_2, \dots, z_d\}$, soit $z_j \in \{z_1, z_2, \dots, z_d\}$
DIMENSIONS & INDEX	
M	Nombre de facteurs dans un vecteur $\boldsymbol{v} = \{v_1, v_2, \dots, v_M\}$
d	Nombre de dimensions d'un code $\boldsymbol{z} = \{z_1, z_2, \dots, z_d\}$
i	Index de facteur, soit $v_i \in \{v_1, v_2, \dots, v_M\}$
j	Index de dimension d'un code, soit $z_j \in \{z_1, z_2, \dots, z_d\}$
AUTRES	
θ	Angle de rotation d'un objet dans une image relativement à un axe géométrique quelconque
C	Nombre de sous-ensembles échantillonnés pour les métriques *intervention-based*

INTRODUCTION

Supposons une tâche de reconnaissance vocale cherchant à prédire l'identité du locuteur à partir d'un enregistrement vocal. La voix de tous les locuteurs est théoriquement unique, et comporte donc une structure harmonique et une musicalité unique. Avant l'arrivée de l'apprentissage profond, il n'était pas recommandé de directement tenter de prédire l'identité du locuteur directement à partir des >8000 échantillons par secondes d'un enregistrement vocal. Plutôt, des ensembles de caractéristiques vocales ciblées étaient extraits tels que l'ensemble GeMAPS (Eyben, et al., 2015), incluant la fréquence fondamentale, l'intensité et plusieurs paramètres représentatifs du contenu spectrale d'une élocution. Ces caractéristiques représentent de manière plus compacte un locuteur comparativement à des échantillons d'enregistrement vocal, et ces caractéristiques sont aussi plus explicites puisque la relation entre la donnée d'entrée et l'identité du locuteur est beaucoup plus simple à définir. Le regroupement des caractéristiques extraites consiste en un code qui est unique au locuteur, et divers codes peuvent être utilisés afin de bâtir un espace de représentation des locuteurs.

Dans le cas des modèles d'apprentissage profond, ceux-ci optimisent d'eux-mêmes une représentation des données adaptée à une tâche spécifique (Mahony, et al., 2019). L'utilisation de l'apprentissage profond permet généralement d'obtenir des codes plus riches en information pour une même dimensionnalité, sans avoir à définir un ensemble précis de caractéristiques a priori.

Bien que les codes soient utiles pour des tâches simples comme la reconnaissance du locateur, ils peuvent aussi être utilisés dans des tâches génératives. Par exemple, la Figure 0.1 illustre une application où un tel code peut adapter le style de locution d'un enregistrement vocal à une phrase de forme textuelle (Wang, et al., 2018).

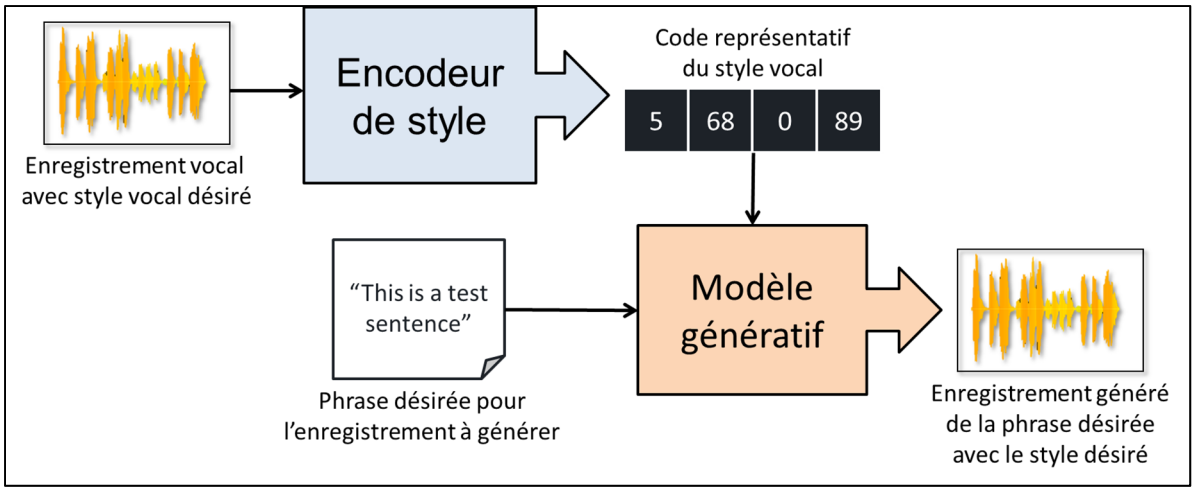

Figure 0.1 Visualisation d'un modèle génératif avec un certain contrôle sur le style vocal

Dans l'exemple de la Figure 0.1 l'ensemble des codes extraits réunis à l'intérieur d'un même nuage de points constitue une représentation, où les styles de différents enregistrements vocaux y sont décrits. Ce modèle est donc un exemple d'apprentissage de représentations optimisé sur une tâche de génération de voix artificielle et stylisée. Il s'agit d'une tâche objective suffisamment générique pour permettre de converger vers une représentation polyvalente. Cela permet l'accomplissement de tâches connexes en aval. Par exemple, une représentation polyvalente pourrait être tout autant utile à la génération de nouveaux exemples, la classification de caractéristiques identifiables et l'agglomération d'observations. Pour évaluer ce niveau de polyvalence, il est trop coûteux d'identifier et tester exhaustivement toutes les tâches connexes possibles. Il est donc préférable d'évaluer la représentation selon une qualité commune à toutes les représentations polyvalentes, soit le démêlage (*disentanglement*).

Le démêlage est défini différemment d'un ouvrage à l'autre (Bengio, Courville, & Vincent, 2013) (Higgins, et al., 2018) (Eastwood & Williams, 2018) (Suter, Miladinović, Schölkopf, & Bauer, 2019) (Ridgeway & Mozer, 2018). Ce document s'intéresse davantage aux définitions offertes par (Eastwood & Williams, 2018) et (Ridgeway & Mozer, 2018). Dans la plupart de ces définitions, le démêlage d'une représentation s'évalue en fonction des facteurs d'un ensemble de données. Un facteur est une variable identifiant les éléments présents dans

chacune des observations d'un ensemble. Par exemple, tel qu'illustré à la Figure 0.2, DSprites (Higgins, et al., 2017) est un ensemble d'images de formes géométriques qui sont placées en fonction de 4 facteurs, soit l'échelle de grandeur, la rotation et la position x/y. Dans ce cas, ces facteurs dits génératifs sont directement responsables de la génération de l'ensemble. Dans d'autres types d'ensembles, les facteurs pourraient être annotés suivant l'observation d'un humain ou l'extraction d'une caractéristique par un algorithme.

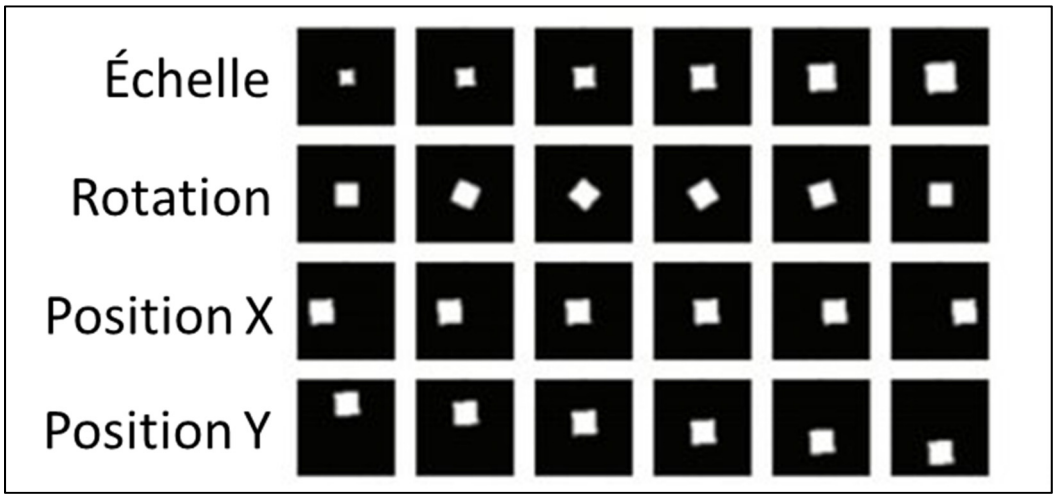

Figure 0.2 Exemples de facteurs tirés de l'ensemble DSprites

Selon (Eastwood & Williams, 2018) et (Ridgeway & Mozer, 2018), le démêlage est défini selon 3 propriétés communes. Premièrement, la **qualité explicite** (*explicitness*) d'une représentation correspond essentiellement à la quantité et la qualité de l'information capturée dans la représentation relativement aux facteurs. Par exemple, une représentation explicite peut être utilisée afin de prédire les facteurs des observations directement à partir des codes extraits. Un exemple de qualité explicite est illustré à la Figure 0.3.

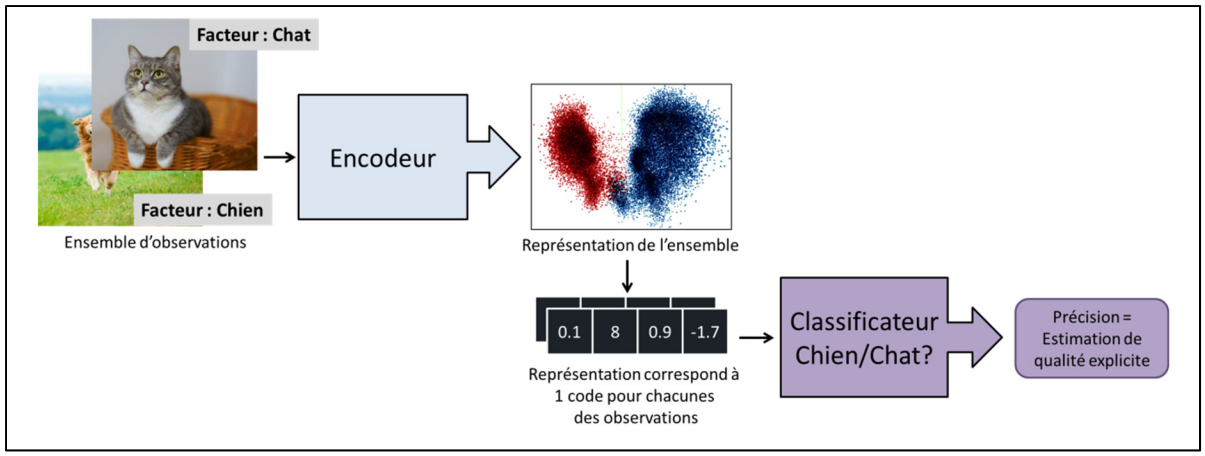

Figure 0.3 Exemple d'évaluation de la qualité explicite d'une représentation

Deuxièmement, la **modularité** (*modularity*) s'intéresse davantage à ce que chacune des dimensions de codes soit représentative d'un seul facteur. L'avantage d'une représentation modulaire consiste en ce que les facteurs potentiellement identifiés par l'encodeur se trouvent dans des sous-ensembles bien définis de dimensions des codes. La Figure 0.4 illustre une telle interprétation, où une ligne mince et une ligne épaisse représentent respectivement une importance faible et forte, et où une cellule rouge et une cellule verte représentent respectivement une dimension de code non modulaire et modulaire. Dans cet exemple, les dimensions des codes z_1 et z_2 sont modulaires puisqu'elles sont seulement représentatives d'un seul facteur. La dimension des codes z_3 n'est pas modulaire, puisqu'elle est représentative de plusieurs facteurs.

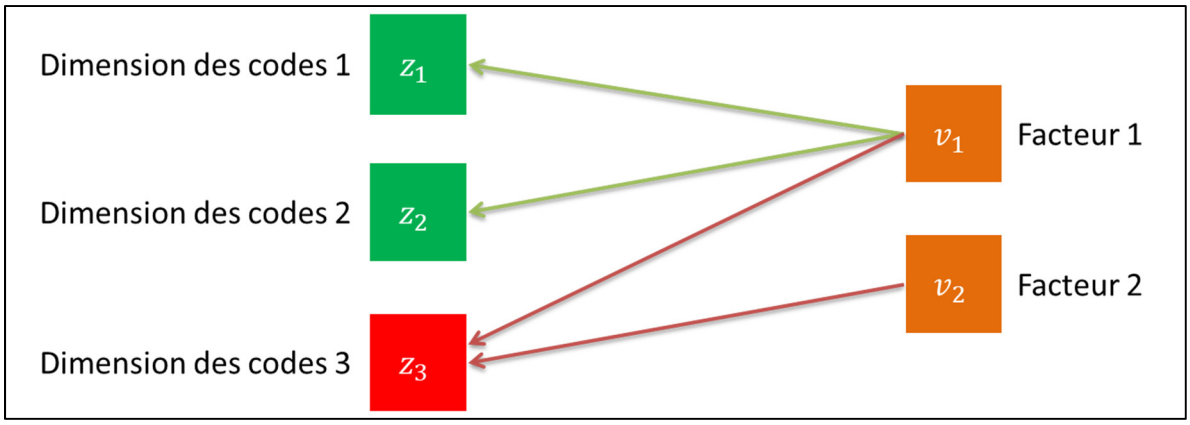

Figure 0.4 Interprétation de modularité et non-modularité d'une représentation

En dernier, la **compacité** (*compactness*) est une propriété cherchant à identifier si un facteur est davantage décrit avec au plus une dimension des codes. La compacité encourage donc des relations univariées entre les codes et facteurs. La Figure 0.5 illustre un exemple d'interprétation de la compacité, où une cellule verte et une cellule rouge définissent respectivement un facteur représenté de façon compacte et non compacte. Dans celui-ci, le facteur v_1 est représenté de façon non compacte contrairement à v_2 qui est représenté de façon compacte, soit par une seule dimension des codes.

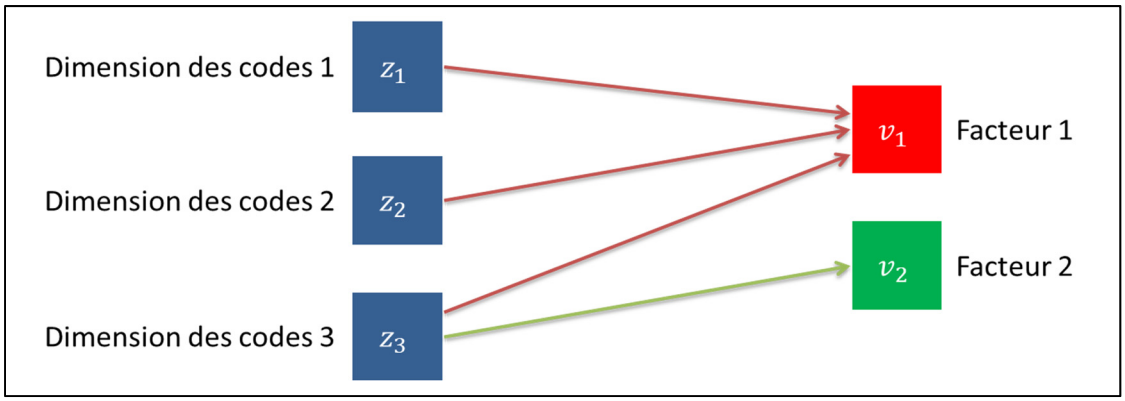

Figure 0.5 Interprétation de compacité et non-compacité de la représentation des facteurs

L'obtention d'une représentation démêlée en fonction des trois propriétés du démêlage décrites est utile pour les cas d'applications de nature générative où on cherche à manuellement contrôler des propriétés difficiles à décrire par l'humain. Par exemple, un praticien pourrait entraîner divers modèles qui fournissent chacun une représentation des données d'entrée. Meilleures sont les mesures de qualité explicite, modularité et compacité par rapport à un facteur, plus facilement que le praticien pourra manuellement manipuler la représentation afin que le modèle associé génère des exemples en fonction de ce facteur. Comparativement à l'exemple de la Figure 0.1, cette méthodologie permettrait de générer de nouvelles voix sans avoir accès au préalable à un enregistrement contenant le style vocal désiré. Autre que la génération d'enregistrements audio, il y existe plusieurs autres domaines d'application potentiels, tels que la génération d'animations automatisée faciales via apprentissage profond (Paier, Hilsmann, & Eisert, 2020), la génération d'images (Li, Singh, Ojha, & Lee, 2020) et la génération de texte (Balasubramanian, Kobyzev, Bahuleyan, Shapiro, & Vechtomova, 2020).

La qualité explicite, modularité et compacité sont des propriétés algorithmiquement quantifiables. Afin de remplacer les méthodes antérieures d'évaluation subjectives du démêlage, des métriques de démêlage quantitatives ont été proposées (Higgins, et al., 2017) (Kim & Mnih, 2018) (Kim, Wang, Sahu, & Pavlovic, 2019) (Kim, Wang, Sahu, & Pavlovic, 2019) (Suter, Miladinović, Schölkopf, & Bauer, 2019) (Eastwood & Williams, 2018) (Kumar, Sattigeri, & Balakrishnan, 2018) (Ridgeway & Mozer, 2018) (Sepliarskaia, Kiseleva, & de Rijke, 2020) (Do & Tran, 2020) (Li, Murkute, Gyawali, & Wang, 2020) (Ridgeway & Mozer, 2018) (Chen, Li, Grosse, & Duvenaud, 2018) (Do & Tran, 2020). Nous dénotons ces métriques comme des métriques de démêlage supervisées, puisqu'elles sont spécialisées à mesurer la qualité représentative relative aux facteurs d'un ensemble de données.

Il existe aussi d'autres métriques mesurant la quantité d'information partagée entre une représentation et un ensemble de données qui ne nécessitent pas de facteurs (Do & Tran, 2020) (Duan, et al., 2020) (Liu, Thermos, Valvano, Chartsias, O'Neil, & Tsaftaris, 2020).

Nous les dénommons comme des métriques non-supervisées. Ceux-ci ne sont pas considérés dans le cadre de cette étude.

Suite à la proposition de diverses métriques de démêlage, des désaccords entre celles-ci quant à l'identification des représentations les plus démêlées mettent en doute leur fiabilité (Locatello, et al., 2019) (Abdi, Abolmaesumi, & Fels, 2019). Dans ces ouvrages, les représentations mesurées sont obtenues à partir de modèles d'apprentissage de représentations non supervisés. Au préalable, la réelle qualité des propriétés mesurées dans de telles représentations est inconnue, ce qui complique l'interprétation des mesures individuelles obtenues. Cette incapacité de vérifier les métriques individuellement implique aussi la difficulté de comparer les résultats obtenus. Alors que (Sepliarskaia, Kiseleva, & de Rijke, 2020) propose une documentation du comportement des métriques, l'étude est cependant limitée à vérifier deux cas, soit l'analyse des valeurs pouvant être retournées par les métriques lorsque les propriétés qu'elles sont supposées mesurer sont absentes ou présentes.

Dans cet ouvrage, nous visons à bonifier cette dernière étude en caractérisant davantage les métriques. Nous proposons d'effectuer cela à l'aide de scénarios représentatifs prédéfinis. Ces scénarios fournissent des représentations générées en fonction de propriétés contrôlables et simples, ce qui a comme objectif de simplifier l'interprétation des résultats obtenus. Ainsi, la difficulté d'évaluer les représentations provenant d'un modèle non supervisé est évitée, puisque les qualités des représentations proposées sont au préalable connues.

Ce document contribue positivement au domaine de l'apprentissage de représentations sur plusieurs points. D'abord, dans le CHAPITRE 2, une nouvelle taxonomie des métriques est définie, regroupant l'ensemble des métriques en 3 familles selon leurs mécanismes utilisés. Il s'agit des familles Intervention-based, Predictor-based et Information-based.

Le CHAPITRE 3 fournit une analyse à jour de la corrélation entre les métriques identifiées par l'auteur jusqu'à maintenant. Les résultats de cette analyse appuient la problématique de désaccord entre les métriques face à l'identification des représentations les plus démêlées.

Le CHAPITRE 4 propose d'adopter des définitions de propriétés désirables exclusives aux métriques, soit la cohérence des mesures relativement aux propriétés mesurées (4.2.1), l'absence s'assomption relationnelle entre les codes et les facteurs (4.2.2), la robustesse des mesures de modularité et compacité à l'induction de bruit (4.2.3), la robustesse des mesures aux dimensions impertinentes de codes (4.2.4) et la stabilité paramétrique des mesures (4.2.5). Avec chacune de ces propriétés, 5 scénarios de tests sont proposés afin de concrètement évaluer ces propriétés. Il s'agit à notre connaissance de la première évaluation empirique du genre sur ces métriques. Finalement, avec les interprétations effectuées il est conclu que la métrique DCI RandomForest retourne les mesures les plus fiables.

Finalement, le CHAPITRE 5 discute de l'impact des résultats obtenus. Plusieurs sources de cas d'erreurs sont discutées, telles que la discrétisation qui est commune à un grand ensemble de métriques. D'abord, les relations entre différentes propriétés du démêlage sont définies. Ensuite, puisque la mesure du démêlage est souvent effectuée sur des ensembles expérimentaux de données, certaines spécificités relatives à l'utilisation d'ensembles réels de données sont soulignées. Des considérations pratiques quant au choix des métriques utilisées par le praticien et la priorisation de certaines propriétés du démêlage sont ensuite suggérées. Finalement, des pratiques transparentes pour rapporter les mesures sont suggérées pour les études futures.

Il est important de noter que la recherche effectuée dans le cadre de ce mémoire fût le fruit d'un travail d'équipe. D'abord, l'étudiant a participé à la rédaction d'un article de forme similaire en tant qu'auteur secondaire (Zaidi, Boilard, Gagnon, & Carbonneau, 2020). L'étudiant a joué un rôle équivalent à un second auteur, ce qui explique pourquoi il n'a pas plutôt rédigé un mémoire par article.

Tout de même, l'étudiant a plusieurs contributions importantes à ce travail. D'abord, dans le cadre du chapitre 1, il a contribué à établir le recueil des métriques en implémentant celles qui n'étaient pas publiquement accessibles en ligne, ainsi qu'à l'identification d'une taxonomie pour les familles de métriques. Ensuite, il a contribué à la récolte et la synthèse des résultats du chapitre 2, ce qui a permis de souligner les désaccords entre les métriques. Dans le cas du chapitre 3 et 4, l'étudiant a initialement obtenu certains résultats pouvant être interprétés similairement à ceux obtenus dans le cadre de l'article, mais admet que les contributions de l'étude n'ont pu être correctement appréciées seulement lorsque des chercheurs plus expérimentés ont révisé en profondeur la méthodologie expérimentale, les propriétés caractérisées ainsi que les scénarios proposés. L'étudiant admet donc que les contributions les plus importantes de cet article sont dues en grande part à ces chercheurs.

Finalement, l'étudiant a opté à présenter dans ce mémoire les résultats en lien avec cette révision plutôt que les résultats initiaux qu'il a lui-même obtenus. Considérant cela, il a fait l'effort de décrire avec davantage de détails comparativement à l'article plusieurs aspects discutés afin de démontrer sa maîtrise du sujet. Le mémoire bonifie donc en quelques sortes l'article qui a été rédigé, que ce soit par l'inclusion de davantage de détails mathématiques et de figures explicatives pour les métriques, un plus grand détail dans l'interprétation des résultats, ou par la présentation d'autres exemples concrets dans le chapitre de discussion.

CHAPITRE 1

COMMENT MESURER LE DÉMÊLAGE?

Avant que les premières métriques soient proposées, le démêlage de la représentation d'un modèle d'apprentissage supervisé était mesuré par exploration manuelle des différentes dimensions de l'espace de représentation (Kingma & Welling, 2013). Cette exploration consiste d'abord à définir un code initial, où plusieurs autres codes sont déterminés avec des bonds fixes dans une ou plusieurs dimensions spécifiques. Ensuite, le modèle génératif habituellement fourni dans le contexte d'apprentissage non supervisé s'occupe de générer une observation associée à chacune des représentations. Par exemple, dans (Higgins, et al., 2017), une représentation de formes géométriques est apprise à partir d'un modèle d'apprentissage de représentations, où différentes dimensions sont démontrés par exploration d'être en relation avec la position (x,y), la taille et la rotation de la forme. Cette relation a été déterminée par exploration de la représentation. Un autre exemple aussi tiré de (Higgins, et al., 2017) est illustrée à la Figure 1.1, soit un exemple d'exploration d'une représentation démêlée d'images de visages.

L'analyse de la qualité de ces explorations a le problème d'être subjective. Il est donc très difficile de déterminer une représentation comme plus démêlée parmi un ensemble. De plus, cette analyse manuelle devrait être effectuée pour chacune des configurations d'optimisation des hyperparamètres d'entraînement. Ainsi, il devient rapidement trop coûteux de comparer des modèles par cette méthode.

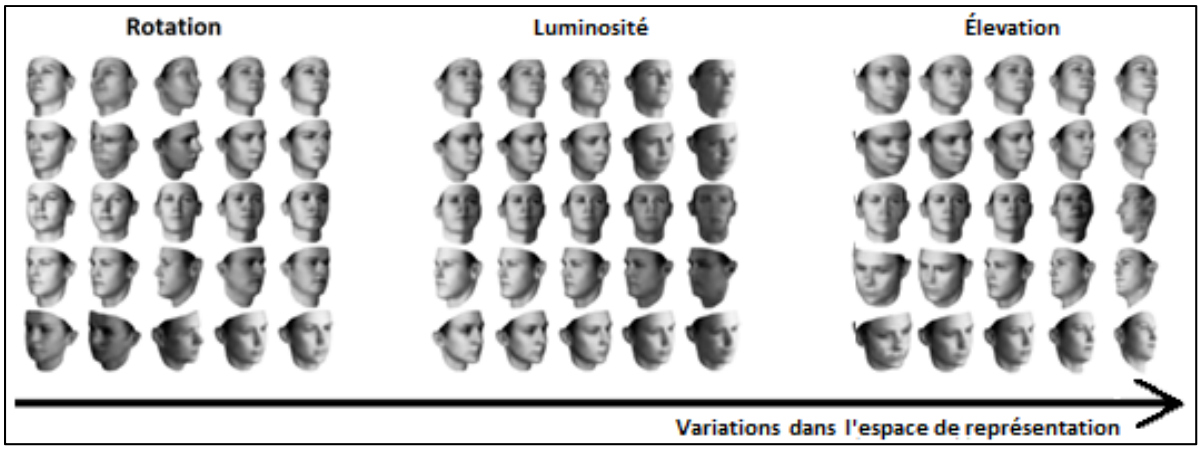

Figure 1.1 Exemple d'exploration dans l'espace de représentation
tiré de Higgins, et al. (2017, p.4)

1.1 Propriétés identifiables du démêlage

Afin d'alléger le processus de comparaison manuelle, des métriques ont été proposées visant la quantification de propriétés mesurables du démêlage. Une première de ces propriétés est la qualité explicite, soit la capacité d'obtenir une représentation capturant suffisamment d'information afin de précisément prédire la valeur d'un facteur. Dans la littérature, cette propriété est nommée *Explicitness* (Ridgeway & Mozer, 2018) ou *Informativeness* (Eastwood & Williams, 2018). La différence principale entre les deux définitions est que celle d'*Explicitness* considère une représentation explicite lorsqu'une relation linéaire entre les codes/facteurs est découverte. Quant à la définition de *Informativeness,* il est souligné que pas tous les facteurs peuvent être représentés linéairement, et considèrent aussi important de déterminer comme explicite une relation non linéaire et multimodale.

La deuxième propriété est nommée modularité. Cette propriété considère qu'afin de faciliter l'interprétation des codes, il est favorable que leurs dimensions individuelles soient explicites d'un minimum de facteurs. Cette propriété est désirable puisque cela permet de facilement identifier et regrouper les dimensions de codes contrôlant un même facteur. Une forte modularité indique donc qu'une dimension des codes est explicite d'un seul facteur, et qu'un

changement de valeurs d'un autre facteur est associé à un changement minimal ou nul dans cette dimension des codes. De plus, considérant que les facteurs sont habituellement indépendants l'un de l'autre, la mesure de la modularité permet de quantifier la réflexion de cette indépendance dans les dimensions des codes représentant ces facteurs. Cette définition est nommée *Disentanglement* dans (Eastwood & Williams, 2018), mais nous optons pour une traduction française tirée de *Modularity* (Ridgeway & Mozer, 2018) afin d'éviter de confondre la propriété avec le sujet général du démêlage (disentanglement).

La troisième propriété est nommée compacité. Afin qu'une représentation soit interprétable, il est discuté par certains auteurs qu'il est important de minimiser le nombre de dimensions de codes décrivant un facteur. Dans la littérature, cette propriété se nomme *compactness* (Ridgeway & Mozer, 2018) ou *completeness* (Eastwood & Williams, 2018). Une représentation avec une compacité maximale indique qu'une seule dimension de code est représentative d'un facteur. Cette propriété permet de trouver des représentations plus simples. Cependant, selon (Ridgeway & Mozer, 2018), la compacité est nuisible pour la représentation de certains facteurs. Un exemple est donné avec un facteur génératif $\theta \in [0^o, 360^o]$. La représentation la plus compacte capable de représenter la discontinuité à $\theta = 360^o = 0^o$ est une fonction multivariée : $\theta \rightarrow [sin\theta,\ cos\theta]$. Favoriser une représentation plus compacte réduit considérablement la potentielle qualité de bien représenter un tel facteur. De plus (Ridgeway & Mozer, 2018) discutent de l'avantage d'une représentation contenant plusieurs solutions représentatives. Cela n'est pas possible à l'intérieur d'une représentation trop compacte. En effet, il est possible qu'une représentation à plusieurs dimensions potentiellement superflues contienne plusieurs solutions afin de représenter un même facteur, certaines potentiellement meilleures que d'autres.

1.2 Métriques de démêlage

Plusieurs métriques supervisées mesurant diverses propriétés du démêlage ont été proposées. Dans cette section, ces métriques sont décrites individuellement. Nous mettons en commun leurs similarités en les rassemblant sous 3 familles. La Figure 1.2 illustre ces familles en plus

d'indiquer les propriétés du démêlage mesurées par ces métriques. De plus, nous définissons comme holistiques (holistic) les métriques capturant plus qu'une propriété.

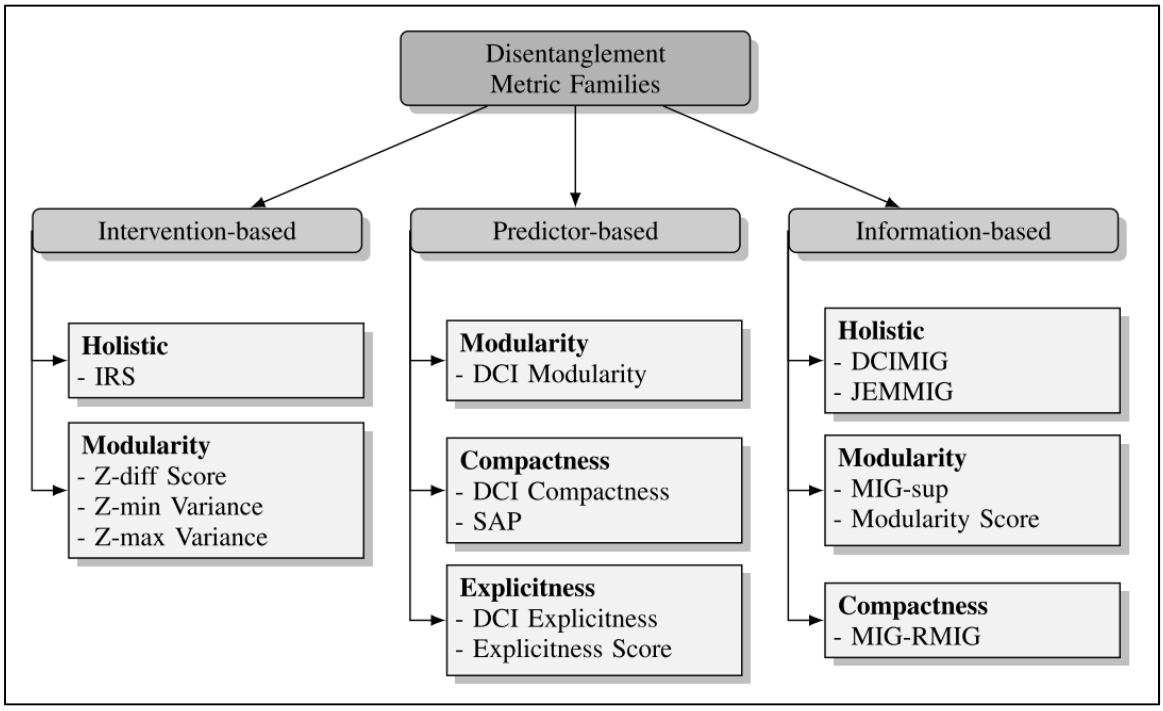

Figure 1.2 Taxonomie des différentes familles de métriques
tiré de Zaidi, Boilard, Gagnon, & Carbonneau (2020, p.5)

Pour décrire les métriques, nous assumons l'utilisation d'un ensemble de données $D^{(N)} = \{V^{(N)}, X^{(N)}, Z^{(N)}\}$ où les facteurs dictent complètement le contenu des observations. Dans cet ensemble, les données correspondant aux mêmes index de $V^{(N)}, X^{(N)}$ et $Z^{(N)}$ sont en relation les uns avec les autres par la chaîne d'opérations $\boldsymbol{z} = r(g(\boldsymbol{v}))$.

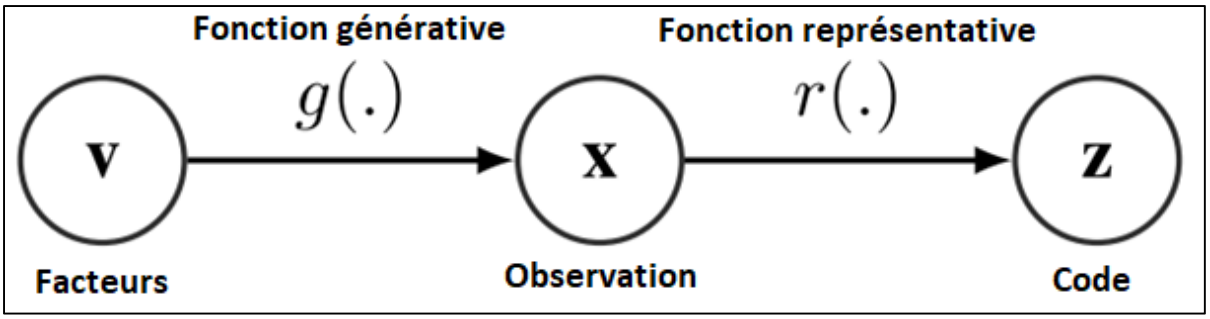

Figure 1.3 Nomenclature des variables utilisées pour décrire les métriques
tiré de Zaidi, Boilard, Gagnon, & Carbonneau (2020, p.5)

Les observations $\boldsymbol{x} \in X^{(N)}$ sont individuellement générées à partir de vecteurs de facteurs $\boldsymbol{v} \in V^{(N)}$. Ces vecteurs contiennent M facteurs génératifs, soit $\boldsymbol{v} = \{v_1, v_2, \dots, v_M\}$. Les codes $\boldsymbol{z} \in Z^{(N)}$ sont individuellement issus d'une fonction de représentation $r(\boldsymbol{x}) = \boldsymbol{z}$, où $\boldsymbol{z} = \{z_1, z_2, \dots, z_d\}$ est un code de dimension d.

En général, la variable i est utilisée pour définir l'index de facteurs v_i et j est définie comme l'index d'une dimension des codes z_j.

1.3 Métriques intervention-based

Supposons la chaîne $\boldsymbol{z} = r\big(g(\boldsymbol{v})\big)$, les métriques ***intervention-based*** mesurent le démêlage en faisant varier la valeur des facteurs afin de quantifier l'influence de telles manipulations sur la valeur des codes. Cette manipulation est suivie d'une évaluation cause-effet. Ce processus est défini comme une intervention dans (Suter, Miladinović, Schölkopf, & Bauer, 2019).

Génériquement, les métriques de cette famille nécessitent d'abord l'échantillonnage d'un sous-ensemble $\mathcal{V}^{(\mathcal{N})} \subseteq V^{(N)}$ en fonction d'une contrainte sur la valeur de certaines dimensions des facteurs $v_i \in \boldsymbol{v} \in \mathcal{V}^{(\mathcal{N})}$. Cette contrainte est différente d'un sous-ensemble échantillonné à l'autre.

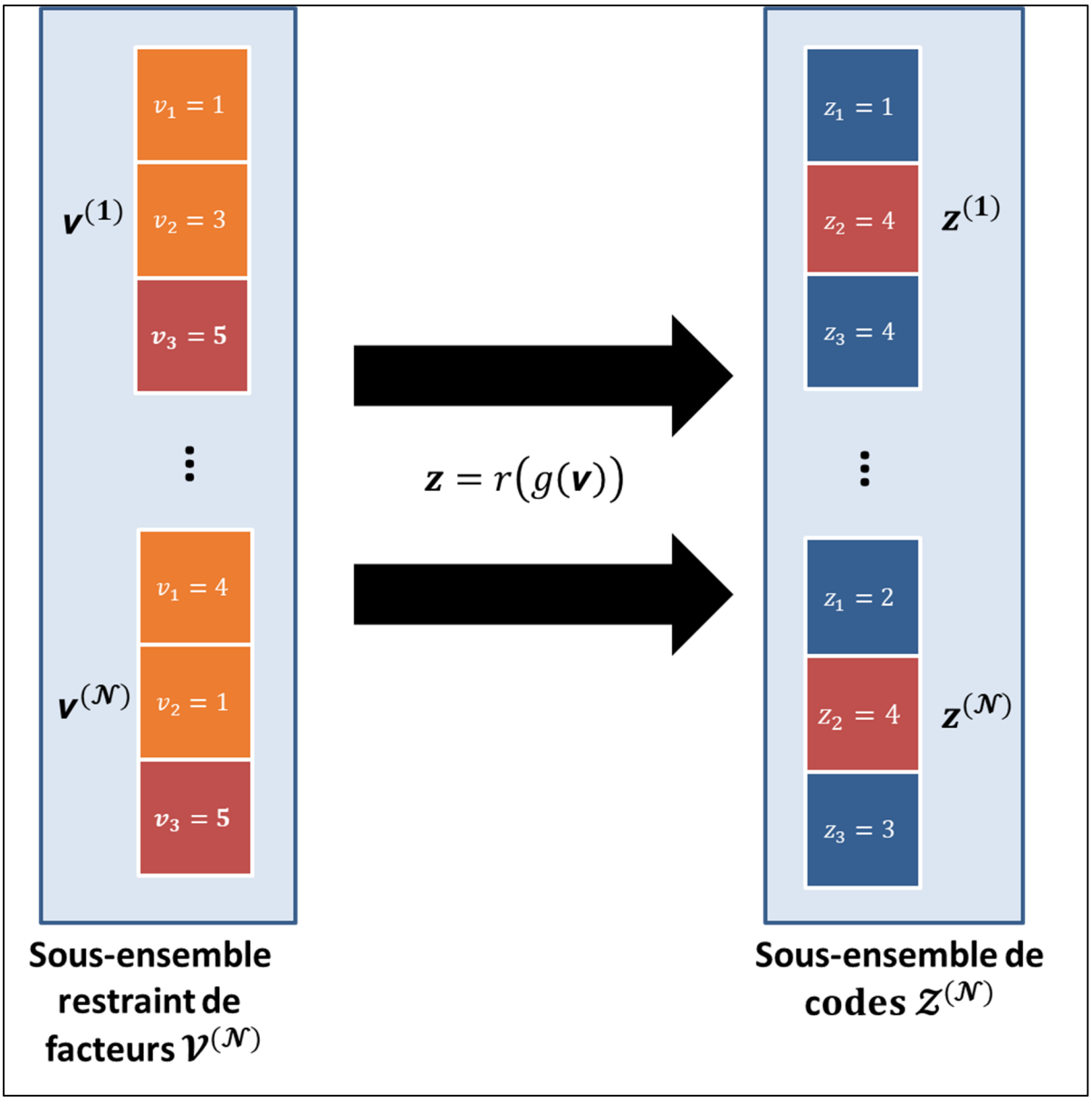

Figure 1.4 Exemple générique d'une manipulation avec un seul sous-ensemble pour les métriques *intervention-based*

Par la chaîne d'opérations $\boldsymbol{z} = r(g(\boldsymbol{v}))$, chacun des ensembles de facteurs $\boldsymbol{v} \in \mathcal{V}^{(\mathcal{N})}$ sont associés à un sous-ensemble de codes $\boldsymbol{z} \in \mathcal{Z}^{(\mathcal{N})}$. Ainsi, la contrainte imposée sur les facteurs dans $\mathcal{V}^{(\mathcal{N})}$ influencera les statistiques de $\mathcal{Z}^{(\mathcal{N})}$. Cela permet donc d'estimer le démêlage entre

$\mathcal{V}^{(\mathcal{N})}$ et $\mathcal{Z}^{(\mathcal{N})}$. La répétition de ce processus permet de déterminer le démêlage global dans la représentation par une agglomération de statistiques locales.

Dans l'exemple de la Figure 1.4, les statistiques de l'ensemble $\mathcal{Z}^{(\mathcal{N})}$ indiquent une variance nulle à la dimension z_2 lorsque les facteurs à l'index $i = 3$ sont restraints à $v_3 = 5$. Cette statistiques pourra donc établir qu'il existe une relation entre v_3 et z_2 dans ce cas spécifique. Il suffira d'observer cette relation avec d'autres sous-ensembles $\mathcal{Z}^{(\mathcal{N})}$ et $\mathcal{V}^{(\mathcal{N})}$ appliquant une contrainte sur le même facteur pour confirmer que cette relation n'est pas une coïncidence.

Un avantage de cette famille de métriques est qu'aucune assomption sur le type de relation entre les codes/facteurs n'est effectuée. Quelques hyperparamètres communs doivent cependant être déterminés avant d'effectuer une mesure, soit la quantité C de sous-ensembles $\mathcal{V}^{(\mathcal{N})}$ échantillonnés et leur population $\mathcal{N}$. De plus, dans le cas de facteurs continus, puisque la contrainte nécessite des facteurs de mêmes valeurs, les métriques nécessiteront que ces facteurs soient discrétisés. Ainsi, un autre hyperparamètre est la granularité de la discrétisation des facteurs continus. Celle-ci doit permettre aux valeurs discrétisées d'être de population suffisamment dense pour être capable de construire les sous-ensembles de population $\mathcal{N}$.

1.3.1 Z-diff

La métrique *Z-diff* (Higgins, et al., 2017), aussi parfois dénommée la métrique β-*VAE*, mesure la modularité d'une représentation. Pour cette métrique, les sous-ensembles de facteurs restreints nécessitent l'échantillonnage de paires. Pour cibler la restriction, un facteur v_i est aléatoirement sélectionné pour chacun des sous-ensembles. La restriction nécessite que toutes les paires contiennent respectivement la même valeur au facteur v_i. Dans la Figure 1.5, un exemple est illustré où la restriction est appliquée sur le facteur v_1.

Ensuite, un sous-ensemble de $\mathcal{N}$ paires de codes $\mathcal{Z}^{(\mathcal{N})} = \{\boldsymbol{z}_a, \boldsymbol{z}_b\}^{(\mathcal{N})}$ correspondant aux paires individuelles du sous-ensemble $\mathcal{V}^{(\mathcal{N})}$ est obtenue**.** Les différences absolues $\boldsymbol{z_{diff}} =$

$|\boldsymbol{z}_a - \boldsymbol{z}_b|$ pour chacune des paires sont obtenues, et la moyenne sur chacune des dimensions de tous les $\boldsymbol{z_{diff}}$ constitue en un point pour un classificateur linéaire cherchant à prédire quel index de facteur i a été fixé. Ce dernier sera entraîné de façon supervisée après C répétitions de l'algorithme. Finalement, la précision de ce classificateur est la mesure retournée par la métrique *Z-diff*. Pour un classificateur aléatoire, le résultat minimum attendu est de $\frac{1}{M}$. Cette valeur est utilisée afin que les mesures soient normalisées entre les bornes [0,1].

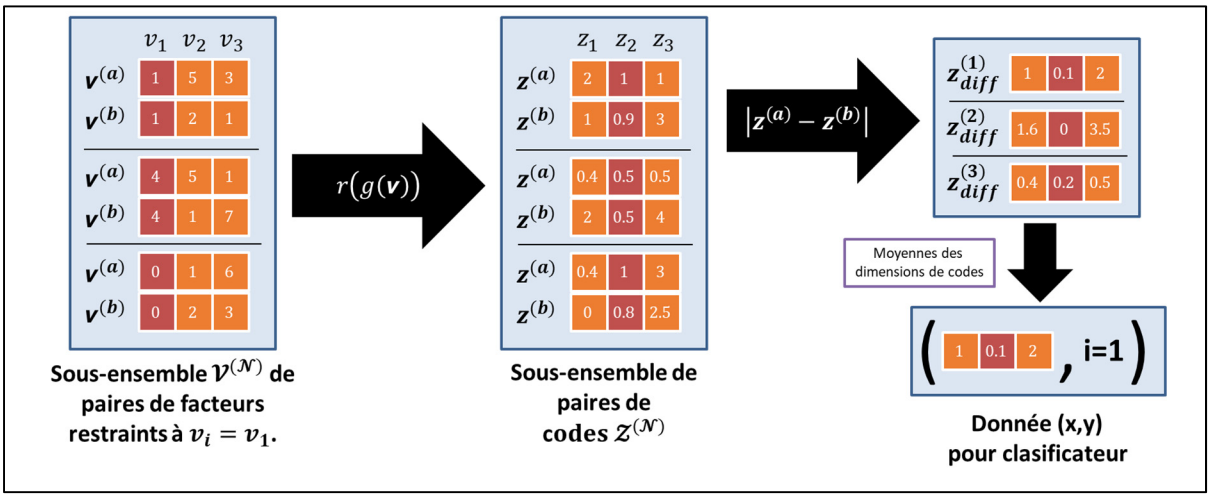

Figure 1.5 Exemple de fonctionnement de la métrique Z-Diff avec un seul sous-ensemble restreint

La logique derrière cette métrique est que les dimensions individuelles de $\boldsymbol{z_{diff}} = |\boldsymbol{z}_a - \boldsymbol{z}_b|$ devraient être près de zéro aux dimensions de codes représentant le facteur v_i fixé, ce qui facilite l'entraînement du classificateur final. Dans l'exemple de la Figure 1.5, la restriction du facteur v_1 cause les codes z_2 à prendre des valeurs très similaires, ce qui cause la donnée envoyée au classificateur à être près de zéro à la même dimension.

Cependant, des cas d'erreurs existent. Premièrement, il est possible que des représentations indésirables reçoivent de bons résultats (Sepliarskaia, Kiseleva, & de Rijke, 2020). En effet, il est possible que 3 paires uniques de 3 dimensions de codes puissent exclusivement

représenter 3 paires uniques de 3 facteurs. Cela cause un cas d'erreur où le classificateur arrive à tout de même reconnaître le facteur fixé et retourner une bonne mesure malgré le chevauchement des relations. L'autre cas d'erreur est identifié dans (Kim & Mnih, 2018). Celui-ci démontre que la métrique *Z-diff* pourrait retourner une bonne mesure dès que tous les facteurs sauf un sont bien représentés.

1.3.2 Z-min Variance

D'abord (Kim & Mnih, 2018) motivent la métrique *Z-min Variance* comme capable de combler les faiblesses de la métrique *Z-diff*. Premièrement (Kim & Mnih, 2018) identifient comme problématique l'optimisation d'un classificateur linéaire dans *Z-diff*, ce qui demande d'établir certains hyperparamètres tels que le choix d'optimisateur et sa configuration, le nombre d'itérations d'entraînement et l'initialisation des poids. De plus, comme mentionné dans la section 1.3.1, *Z-diff* contient des cas d'erreurs critiques motivant la proposition d'une nouvelle métrique. La Figure 1.6 illustre le fonctionnement de cette métrique.

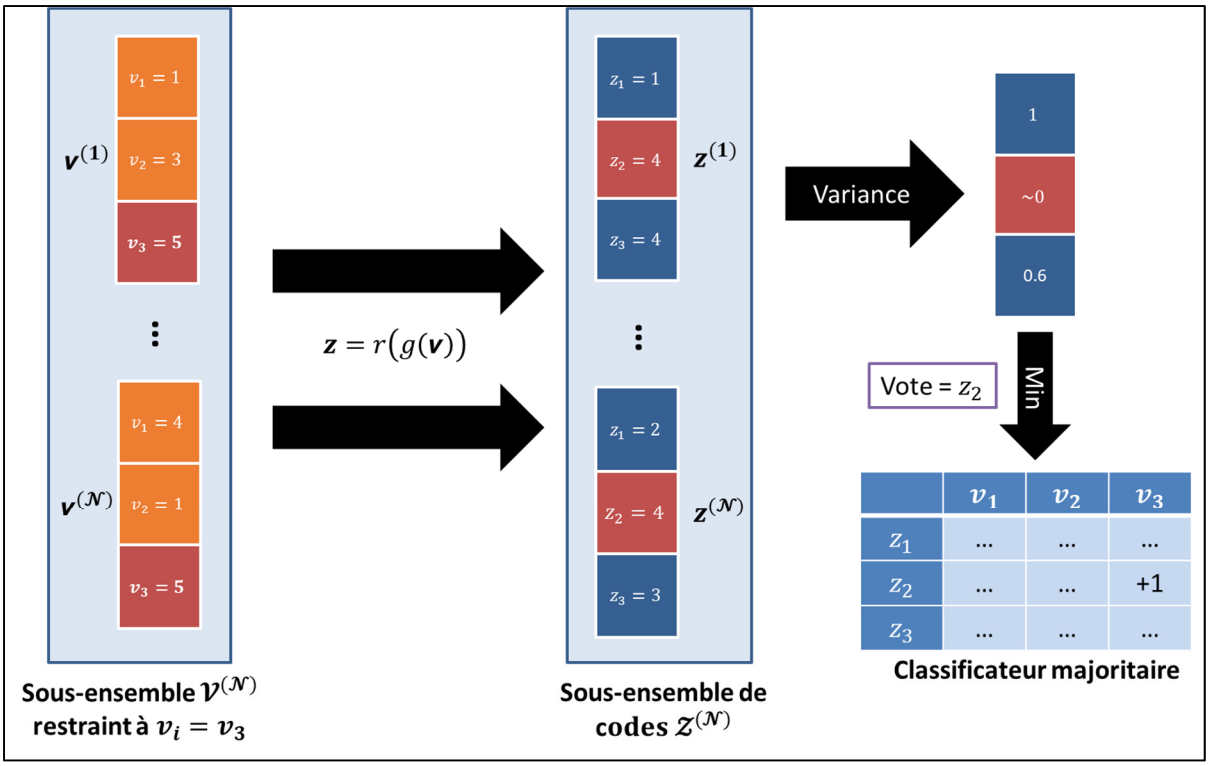

Figure 1.6 Exemple de fonctionnement de la métrique Z-min Variance

Pour obtenir une mesure, un facteur v_i est sélectionné afin d'être fixé. Ensuite, les sous-ensembles $\mathcal{V}^{(\mathcal{N})}$ et $\mathcal{Z}^{(\mathcal{N})}$ sont échantillonnés en respectant une contrainte où tous les facteurs associés $\boldsymbol{v} \in \mathcal{V}^{(\mathcal{N})}$ contiennent une même valeur au facteur génératif $v_i \in \boldsymbol{v}$. Les variances des dimensions des codes de $\mathcal{Z}^{(\mathcal{N})}$ sont calculées et normalisées en fonction de la variance des dimensions des codes de $Z^{(N)}$. La dimension des codes de $\mathcal{Z}^{(\mathcal{N})}$ avec la plus petite variance normalisée est considérée comme la plus fortement associée au facteur fixé v_i. Ce processus est répété C fois. Les C associations sont envoyés comme points pour un classificateur majoritaire. Finalement, le résultat retourné par cette métrique est la précision moyenne de ce classificateur. Pour un classificateur aléatoire, le résultat minimum attendu est de $\frac{1}{M}$. Cette valeur est utilisée afin que les mesures soient normalisées entre les bornes [0,1].

1.3.3 Z-max Variance

La métrique *Z-max Variance* (Kim, Wang, Sahu, & Pavlovic, 2019) quantifie le démêlage similairement à *Z-min Variance*. La logique de *Z-max Variance* est que la dimension d'un code z_j représentatif d'un facteur v_i devrait avoir une forte variance dans le sous-ensemble $\mathcal{Z}^{\mathcal{N}}$ lorsque v_i est le seul facteur dont la valeur est variée. Les étapes sont les mêmes que celles décrites pour Z-min Variance, autres que la sélection d'un facteur génératif à l'index i comme le seul varié plutôt que fixé et les points fournis au classificateur majoritaire étant collectés afin de calculer la variance normalisée maximale au lieu de minimale.

1.3.4 IRS

IRS (Interventional Robustness Score) est une métrique issue de la théorie interventionnelle du démêlage présentée dans la même publication (Suter, Miladinović, Schölkopf, & Bauer, 2019). Cette métrique mesure l'impact de la variation de facteurs nuisibles sur les dimensions de codes déterminées en forte relation avec les facteurs causaux.

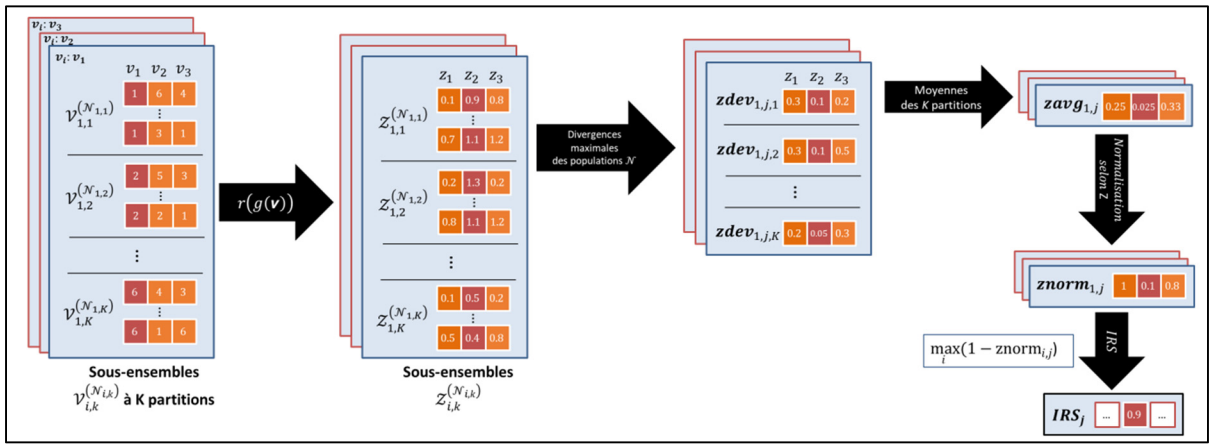

Figure 1.7 Exemple de fonctionnement de la métrique IRS

La Figure 1.7 illustre le fonctionnement de la métrique IRS décrite ci-dessous. Pour effectuer une mesure, une première intervention est préparée en définissant comme causal un facteur à l'index i et les autres comme nuisibles. L'ensemble de données est partitionné en K

ensembles pour chacune des valeurs uniques du facteur $v_i : \mathcal{D}_k = \left\{\mathcal{V}_{i,k}^{(\mathcal{N}_{i,k})},\ \mathcal{Z}_{i,k}^{(\mathcal{N}_{i,k})}\right\}$, où $k \in \{1,2,\dots,K\}$ et $\mathcal{N}_{i,k}$ est le nombre d'éléments dans la partition k du facteur v_i. Pour chacune des $k \in \{1,2,\dots,K\}$ partitions, la déviation maximale individuelle des dimensions des codes de $\mathcal{Z}_{i,k}$ est calculée.

$$\mathrm{zde}v_{i,j,k} = \max_{n^* \in \{1,\dots,\mathcal{N}_{i,k}\}} \left| \left(\mathcal{Z}_{i,k}^{(n^*)}\right)_j - \frac{1}{\mathcal{N}_{i,k}} \sum\nolimits_{n=1}^{\mathcal{N}_{i,k}} \left(\mathcal{Z}_{i,k}^{(n)}\right)_j \right| \tag{1.1}$$

La déviation maximale moyenne de $\mathcal{Z}_{i,k}$ de chacune des dimensions de codes est définie par :

$$\mathrm{zavg}_{i,j} = \frac{1}{K} * \sum_{k=1}^{K} \mathrm{zde}v_{i,j,k} \tag{1.2}$$

L'effet de la manipulation de la valeur des facteurs est finalement observé après avoir pondéré zavg_j selon les déviations maximales des dimensions de codes de $Z^{(N)}$.

$$\mathrm{znorm}_{i,j} = \frac{\mathrm{zavg}_{i,j}}{\max\limits_{n^* \in \{1,\dots,N\}} \left| Z^{(n^*)} - \frac{1}{N} \sum_{n=1}^{N} Z^{(n)} \right|_{\mathrm{j}}} \tag{1.3}$$

Ce dernier résultat indique la déviation maximale moyenne des dimensions de codes individuelles lorsque les facteurs nuisibles sont variés. Ainsi, une forte relation d'une dimension de code z_j avec un facteur causal v_i est indiquée par une faible valeur $\mathrm{znorm}_{i,j}$. L'algorithme est répété pour chacune des valeurs possibles de $i \in \{1,\dots,M\}$. Une mesure IRS est finalement obtenue pour chacune des dimensions de codes :

$$\mathrm{IRS}_j = \max_i (1 - \mathrm{znorm}_{i,j}). \tag{1.4}$$

IRS$_j$ est donc une mesure de la divergence de la dimension de code z_j lorsque des facteurs nuisibles sont variés. Une mesure globale est obtenue en effectuant une moyenne avec chacune des mesures IRS_j obtenues.

À noter que la version présentée ci-dessus est une version simplifiée aussi tirée de la publication originale. La version non simplifiée permet de sélectionner plus que 1 facteur causal, d'ignorer certains facteurs dans les interventions effectuées ainsi que de fournir une mesure unique à des combinaisons de dimensions de codes. Cependant, elle ne propose pas de techniques servant à optimiser les combinaisons. Cette version est celle aussi utilisée pour l'obtention de mesures obtenues dans la publication d'*IRS*.

1.4 Métriques predictor-based

Les métriques ***predictor-based*** évaluent directement l'utilité d'une représentation sur une tâche connexe de prédiction des facteurs. Ces métriques entraînent d'abord des régresseurs/classificateurs afin de prédire les facteurs à partir des codes. La qualité des résultats sur un ensemble de test consiste généralement en une estimation de la qualité explicite de la représentation. Sinon, pour l'obtention d'une mesure de compacité et de modularité, les paramètres internes du modèle entraîné sont algorithmiquement analysés afin de les interpréter similairement aux exemples de modularité et compacité de Figure 0.4 et Figure 0.5.

Ces métriques sont bien équipées pour déterminer la qualité explicite d'une représentation. Par la sélection du modèle interne, ils peuvent être utilisés pour prédire des facteurs continus ou discrets. La variété de modèles possibles signifie cependant qu'ils favoriseront des relations codes/facteurs adaptées au modèle sélectionné à être entraîné pour une métrique donnée. Le plus grand désavantage de ces métriques est qu'elles contiennent beaucoup d'hyperparamètres comparativement aux autres familles de métriques.

1.4.1 Disentanglement, Completeness and Informativeness (DCI)

DCI (Eastwood & Williams, 2018) est une métrique qui retourne une mesure individuelle pour chacune des propriétés de modularité, compacité et de qualité explicite, respectivement nommées par les auteurs *disentanglement*, *completeness* et *informativeness*. La mesure de ces propriétés nécessite d'abord l'entraînement d'un régresseur linéaire avec régularisation Lasso ou d'un régresseur forêt aléatoire (RandomForest) servant à prédire les facteurs. L'utilisation de la variante linéaire privilégiera les représentations linéairement interprétables, lors que la variante forêt aléatoire considère les relations non-linéaires plus complexes. Certains hyperparamètres clés sont automatiquement optimisés à l'entrainement par validation croisée, soit la pénalité α pour le régularisateur Lasso, et la profondeur maximale des arbres pour le régresseur forêt aléatoire.

Dans le cas de la variante utilisant un régresseur linéaire, l'entraînement nécessite que les valeurs des dimensions de codes et les valeurs du facteur à prédire soient normalisées afin qu'elles soient ramenées à des moyennes nulles et des variances unitaires. L'importance R_{ij} de la dimension des codes z_j à la prédiction du facteur v_i est la valeur absolue du poids appris. Les poids appris normalisés par la norme des poids représentent l'importance relative R_{ij} de la dimension de code z_j à la prédiction du facteur v_i.

Dans le cas de l'entraînement d'un régresseur forêt aléatoire, les importances sont contenues dans les séparations optimisées des arbres de décision entraînés. Plus précisément, le nombre de séparations effectuées sur une dimension de code particulière normalisée par le nombre total de séparations détermine l'importance relative R_{ij} de cette dimension des codes z_j à la prédiction d'un facteur v_i. Cette méthode est basée sur l'importance « Gini », qui est utilisée à l'interne des arbres de décisions afin d'optimiser leurs précisions (Breiman, Friedman, Olshen, & Stone, 2017).

1.4.1.1 DCI – Explicitness

Suite à l'entraînement, une estimation de la qualité explicite d'une représentation peut être obtenue. Une fonction de distance (p. ex. l'erreur quadratique moyenne) entre les facteurs de l'ensemble de test et les facteurs prédits est rapportée. Afin de normaliser les résultats, nous définissons qu'une représentation n'est pas explicite lorsqu'une mesure de distance est supérieure à celle théoriquement retournée par l'erreur quadratique moyenne entre deux variables indépendantes et uniformément distribuées. Dans le cadre de nos expériences, toutes les valeurs de codes et facteurs sont normalisées entre [0,1]. Ainsi, pour *DCI Explicitness*, on peut considérer comme au moins légèrement explicite toute erreur quadratique moyenne inférieure à $E[(X-Y)^2] = 1/6$. L'estimation normalisée de la qualité explicite d'une représentation pour un seul facteur est donc :

$$\mathrm{DCI}_{\mathrm{explicitness}} = 1 - 6 * MSE \tag{1.5}$$

Les mesures inférieures à zéro sont ajustées à zéro. La mesure finale de qualité explicite est la moyenne des distances normalisées pour la prédiction de chacun des facteurs.

1.4.1.2 DCI – Compactness

Afin d'estimer la compacité d'une représentation avec DCI, les paramètres des modèles internes entraînés sont analysés afin de déterminer l'importance relationnelle R_{ij} entre chacune des dimensions de codes et chacun des facteurs. Calculer la compacité représentative d'un facteur v_i nécessite d'abord de générer une matrice de probabilité relationnelle p_{ij}, qui est obtenue en normalisant l'importance R_{ij} selon la somme des importances prédictives d'un seul facteur :

$$p_{ij} = R_{ij} / \sum_{k=1}^{d} R_{ik} \tag{1.6}$$

Ensuite, un résultat de *compacité* C_i pour chacun des facteurs sont obtenus par une équation équivalente à l'estimation de l'entropie d'une distribution. Celle-ci quantifie la dominance relative de l'importance d'une dimension de code pour la prédiction d'un facteur :

$$C_i = 1 + \sum_{j=1}^{d} p_{ij} \log_d p_{ij} \tag{1.7}$$

La mesure de compacité finale est la moyenne des mesures C_i obtenues pour chacun des facteurs v_i.

1.4.1.3 DCI – Modularity

Afin d'estimer la modularité d'une représentation avec DCI, il faut d'abord définir la matrice d'importance R_{ij}. Une matrice de probabilité relationnelle p_{ij} est encore nécessaire, mais cette fois où les importances associées à une dimension des codes sont normalisées selon son importance avec tous les facteurs évalués:

$$p_{ij} = R_{ij} / \sum_{k=1}^{M} R_{kj} \tag{1.8}$$

Ensuite, une mesure de modularité D_j est obtenue pour chacune des dimensions de codes par :

$$D_j = 1 + \sum_{i=1}^{M} p_{ij} \log_M p_{ij} \tag{1.9}$$

Afin d'obtenir une mesure globale, il ne reste qu'à pondérer D_j en fonction de la quantité d'information que les dimensions des codes contiennent. La mesure finale est une moyenne pondérée de D_j :

$$DCI_{mod} = \sum_{j=1}^{d} \rho_j D_j \tag{1.10}$$

Le terme de pondération ρ_j mesure l'importance globale relative de chacune des dimensions de codes. Celui-ci permet en plus d'éliminer les dimensions de codes inactives où contenant de l'information représentant d'autres facteurs non-identifiés.

$$\rho_j = \frac{\sum_{i=1}^{M} R_{ij}}{\sum_{k=1}^{d} \sum_{i=1}^{M} R_{ik}} \tag{1.11}$$

1.4.2 Explicitness Score

La *Explicitness Score* (Ridgeway & Mozer, 2018) mesure la qualité explicite d'une représentation à l'aide de la courbe ROC de plusieurs classificateurs logistiques. Plus précisément, des classificateurs binaires sont entraînés dans une configuration « un-contre-tous » (One-Vs-Rest) pour chacune des valeurs d'un facteur, et ce même si celui-ci est continu. Ainsi, une courbe est obtenue pour chacun des classificateurs, et l'aire sous chacune des courbes *(Area Under Curve / AUC)* constitue une estimation de la qualité explicite de la représentation spécifique à une valeur de facteur. Un exemple de la méthodologie ROC-AUC est illustrée à la Figure 1.8.

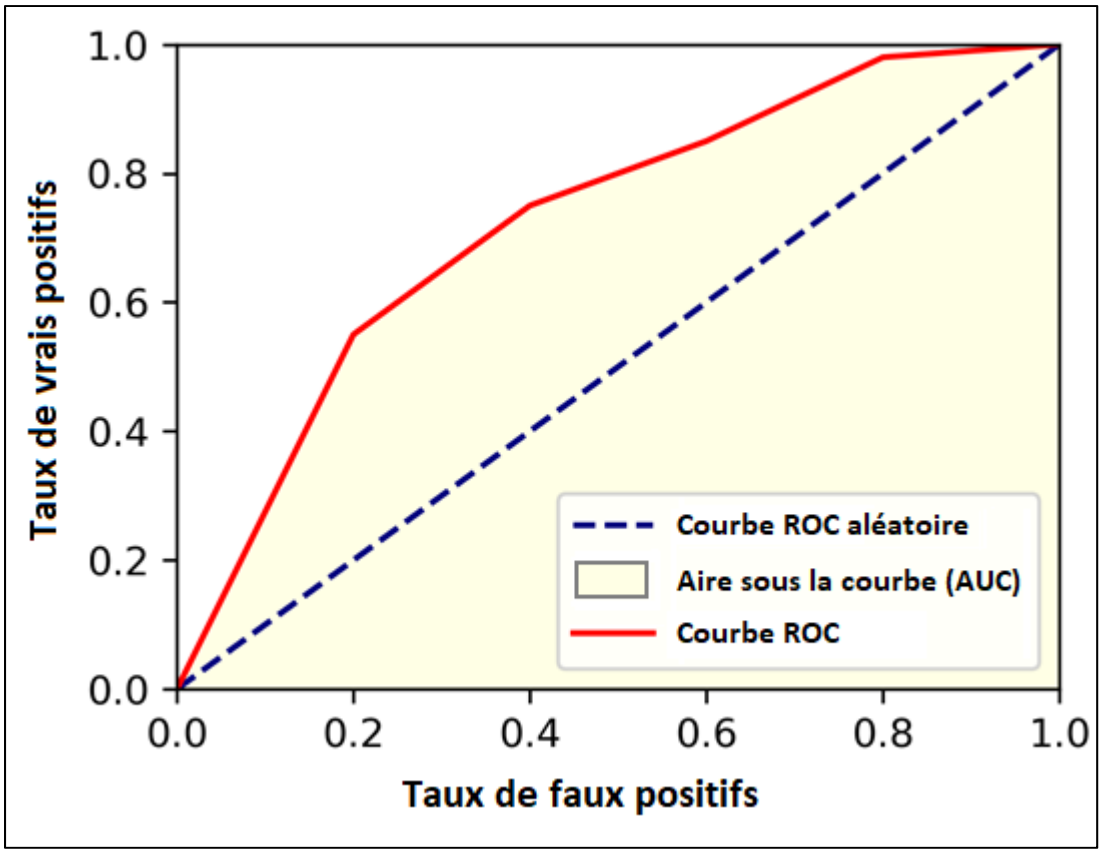

Figure 1.8 Visualisation d'une courbe ROC et de l'aire sous la courbe (AUC)[1]

La moyenne des aires attribuées aux différentes valeurs d'un même facteur constitue la *Explicitness Score* globale de ce facteur. Pour obtenir une mesure davantage globale, la moyenne des mesures globales de facteurs est effectuée.

1.4.3 Separated Attribute Predictability Score (SAP)

Les auteurs de SAP (Kumar, Sattigeri, & Balakrishnan, 2018) observent que les mesures de métriques antérieures ne sont pas corrélées à leurs impressions subjectives du démêlage par

[1] https://towardsdatascience.com/a-simple-explanation-of-the-roc-curve-and-auc-64db32d75541
(Consulté le 1er Décembre 2022)

exploration manuelle dans l'espace de représentation. SAP est donc proposée afin de fournir une mesure mieux corrélée à ces impressions qualitatives.

Afin d'obtenir une mesure avec SAP, une matrice de représentativité R_{ij} est d'abord calculée. Celle-ci enregistre la capacité de décrire un facteur v_i à l'aide d'une seule dimensions des codes z_j. Dans le cas de facteurs continus, R_{ij} est calculée avec le coefficient de détermination R^2, soit une estimation la qualité prédictive d'une régression linéaire entre un seul facteur v_i et une seule dimensions des codes z_j. Dans le cas de facteurs discrets, R_{ij} constitue la précision d'un classificateur linéaire à prédire v_i à l'aide d'une seule dimension de codes z_j.

Suite à l'obtention de R_{ij}, la mesure SAP peut être calculée. Pour chacun des facteurs v_i, les deux dimensions de codes z_* et $z_\circ$ représentant mieux le facteur sont extraits. La différence entre $R_{i*} - R_{i\circ}$ constitue en la mesure SAP relative à un facteur v_i. L'équation suivante démontre ce processus appliqué sur chacun des facteurs afin d'obtenir une mesure globale :

$$\text{SAP} = \frac{1}{\text{M}} \sum_{i}^{M} (R_{i*} - R_{i\circ}) \tag{1.12}$$

Où M correspond au nombre de facteurs génératifs, l'index de dimension des codes $j = *$ à celle la plus fortement associée au facteur v_i, alors que $j = \circ$ correspond à celle la deuxième plus fortement associée au même facteur.

1.5 Métriques information-based

La dernière famille, soit les métriques ***information-based,*** mesurent le démêlage avec des estimations de l'information mutuelle. L'information mutuelle consiste en une mesure de la dépendance statistique entre deux variables aléatoires. Pour ces métriques, l'information mutuelle entre chacune des dimensions de codes et facteurs est d'abord estimée. L'équation suivante est une méthode commune d'estimation de l'information mutuelle :

$$I(v_i, z_j) = \sum_{b_v=1}^{B_{v_i}} \sum_{b_z=1}^{B_z} P(b_v, b_z) \log\left(\frac{P(b_v, b_z)}{P(b_v)P(b_z)}\right) \quad (1.13)$$

Où B_{v_i} et B_z correspondent au nombre de classes discrètes possibles respectives des facteurs individuels et des codes suivant leur discrétisation. B_{v_i} peut varier d'un facteur à l'autre si les facteurs sont catégoriques. Sinon, les facteurs continus sont discrétisés en une même quantité de classes. $P(b_v)$ correspond à la probabilité d'occurrence de la classe b_v. $P(b_z)$ correspond à la probabilité d'occurrence de la classe b_z. $P(b_v, b_z)$ correspond à la probabilité de co-occurrence entre une classe de facteur et une classe d'une dimension de code.

Les métriques de cette famille nécessitent la discrétisation des codes si ceux-ci sont continus. De plus, tel que les métriques intervention-based, l'information mutuelle ne pose aucune assomption sur le type de relation entre les codes/facteurs tel que décrit dans (Do & Tran, 2020).

1.5.1 Mutual information Gap (MIG) / RMIG (Robust MIG)

MIG (Chen, Li, Grosse, & Duvenaud, 2018) est une métrique mesurant la différence (*Gap*) d'information mutuelle des deux dimensions de codes démontrant la plus grande représentativité d'un facteur. MIG est mesuré par l'équation suivante :

$$MIG_i = \frac{I(v_i, z_*) - I(v_i, z_\circ)}{H(v_i)} \quad (1.14)$$

$I(v_i, z_*)$ et $I(v_i, z_\circ)$ correspondent respectivement à la meilleure et deuxième meilleure information mutuelle entre le facteur v_i et les dimensions des codes, et $H(v_i)$ correspond à l'entropie des valeurs de v_i. Cette entropie est utilisée afin d'ajuster les mesures d'information mutuelle entre [0,1]. MIG est calculée pour chacun des facteurs v_i, et la

mesure finale est obtenue par la moyenne des mesures MIG_i obtenues pour chacun des facteurs v_i.

RMIG est proposée dans (Do & Tran, 2020). Il s'agit d'une métrique similaire à MIG avec une formulation différente de l'information mutuelle qui n'est pas utile à nos expériences. Supposons le cas d'apprentissage non supervisé tel qu'illustré à la Figure 1.9. Dans le cas de RMIG, les facteurs et les codes sont issues respectivement des observations. RMIG estime donc l'information mutuelle de façon davantage compatible avec les ensembles de données non expérimentaux. MIG assume plutôt que les codes soient issus des observations, qui elles sont à leur tour issues des facteurs. MIG estime donc l'information mutuelle en considérant que les données proviennent d'un processus génératif, ce qui est compatible avec nos scénarios proposés.

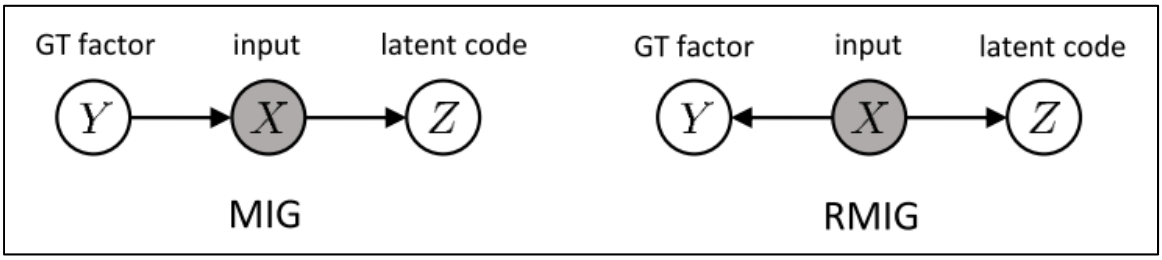

Figure 1.9 Comparaison des assomptions probabilistiques entre MIG et RMIG
tiré de Do & Tran (2020, p.6)

1.5.2 MIG-supplement (MIG-sup)

MIG-sup est une métrique motivée par l'incapacité de MIG de pénaliser une mesure lorsque plusieurs facteurs génératifs sont représentés dans une seule dimension des codes (Li, Murkute, Gyawali, & Wang, 2020). MIG-sup se différencie de MIG en mesurant plutôt la différence d'information mutuelle entre les deux facteurs les plus représentés par une seule dimension de code, tel que définie par l'équation suivante :

$$\text{MIGsup}_j = \frac{I(z_j, v_*)}{H(v_*)} - \frac{I(z_j, v_\circ)}{H(v_\circ)} \tag{1.15}$$

Où $I(z_j, v_*)$ et $I(z_j, v_\circ)$ correspondent respectivement à la meilleure et deuxième meilleure information mutuelle entre une dimension des codes et l'ensemble des facteurs. Les termes $H(v_*)$ et $H(v_\circ)$ sont l'entropie respective à ces deux facteurs, et sont utilisées afin de normaliser les informations mutuelles entre les bornes [0,1]. Une mesure est obtenue pour chacune des dimensions de code z_j, et le résultat global est obtenu par la moyenne de ces mesures.

1.5.3 Modularity Score

Les auteurs de *Modularity Score* (Ridgeway & Mozer, 2018) proposent une métrique basée sur l'information mutuelle capable de quantifier la modularité pour plus que deux facteurs représentatifs d'une dimension de code, contrairement à MIG-sup. La *Modularity Score* mesure l'information mutuelle la plus forte entre une dimension de code et un facteur, et la pondère selon l'information mutuelle de cette même dimension de code relativement aux autres facteurs par l'équation suivante:

$$\text{ModularityScore}_j = 1 - \frac{\sum_{i \in v_{i \neq i_*}} I(v_i, z_j)^2}{(M-1) * I(v_{i_*}, z_j)^2} \tag{1.16}$$

Où i_* correspond à l'index de facteur avec l'information mutuelle la plus forte relativement à la dimension de code z_j évaluée. Une mesure est obtenue pour chacune des dimensions des codes, et leur moyenne est effectuée afin d'obtenir une mesure globale finale.

1.5.4 DCIMIG

DCIMIG (Sepliarskaia, Kiseleva, & de Rijke, 2020) est une métrique mesurant la modularité similairement à MIG-sup. La modularité est une propriété normalement mesurée pour chacune des dimensions de codes. DCIMIG fourni plutôt une mesure de modularité pour

chacun des facteurs en agglomérant les dimensions de codes représentant le plus fortement un même facteur.

Pour effectuer une mesure avec DCIMIG, la différence d'information mutuelle G_j est calculée pour chacune des dimensions de codes z_j :

$$G_j = I(v_*, z_j) - I(v_\circ, z_j) \tag{1.17}$$

Où v_* et $v_\circ$ sont respectivement les facteurs ayant la meilleure et deuxième meilleure information mutuelle avec une dimension de code z_j. Ensuite, les différences G_j sont partitionnées en M groupes P_i, où $i \in \{1,2,\dots,M\}$ et où les différences $G_j \in P_M$ d'une même partition sont calculées avec un même facteur v_*. Dans l'équation suivante, les différences maximales respectives à chacune de ces partitions sont récupérées afin de mesurer le démêlage selon DCIMIG :

$$\text{DCIMIG} = \frac{\sum_{i=1}^{M} \max P_i}{\sum_{i=1}^{M} H(v_i)} \tag{1.18}$$

Dans cette équation, l'entropie des facteurs $H(v_i)$ est utilisée pour normaliser la mesure entre [0,1] en pondérant selon la quantité d'information à capturer dans chacun des facteurs. Si aucune dimension de code ne représente davantage un facteur v_i quelconque, alors $\max P_i = 0$.

Cette métrique favorisera les représentations où l'entièreté de chaque facteur est décrite dans une dimension de code distincte, et n'appliquera pas de pénalités lorsqu'un facteur est aussi décrit partiellement dans d'autres dimensions de code.

1.5.5 Joint Entropy minus Mutual information Gap (JEMMIG)

JEMMIG (Do & Tran, 2020) bonifie la métrique MIG en mesurant aussi à quel point l'information contenue dans une dimension de code est uniquement attribuée au facteur évalué. JEMMIG est calculée ainsi :

$$JEMMIG_i = H(v_i, z_\star) - I(v_i, z_*) + I(v_i, z_\circ) \tag{1.19}$$

Où $H(v_i, z_\star)$ correspond à l'entropie entre v_i et la dimension des codes z_* la plus représentative de ce facteur, $I(v_i, z_*)$ correspond plutôt à l'information mutuelle entre ces mêmes composants et $I(v_i, z_\circ)$ correspond à l'information mutuelle entre v_i et la deuxième dimension de code $z_\circ$ la plus représentative de ce facteur. Tout d'abord, cette équation est calibrée différemment des autres métriques, soit une bonne mesure JEMMIG correspond à une valeur retournée de 0.

Quant au fonctionnement de la métrique, la soustraction $H(v_i, z_\star) - I(v_i, z_*)$ permet de vérifier si l'information présente dans z_* est uniquement attribuée au facteur v_i. Il s'agit d'une vérification indirecte de la modularité des dimensions de codes évaluées. De plus, afin de favoriser les représentations compactes, le terme $(v_i, z_\circ)$ pénalise la métrique si non nulle. Le résultat final correspond à une moyenne des mesures $JEMMIG_i$ obtenues pour chacun des facteurs v_i. À noter que la version implémentée dans cette étude est recalibrée entre [0,1] où une mesure de 0 correspond à une mauvaise mesure et 1 une bonne mesure, soit comme les autres métriques afin de simplifier les interprétations.

CHAPITRE 2

ÉVALUATION DES DÉSACCORDS ENTRE LES MÉTRIQUES

Quelques études quantifient la corrélation entre différentes mesures de métriques (Locatello, et al., 2019) (Sepliarskaia, Kiseleva, & de Rijke, 2020) et mettent en évidence les difficultés à définir le modèle d'apprentissage générant les représentations les plus démêlées. Dans ce chapitre, nous effectuons une analyse de corrélation entre les métriques similaire à celles citées ci-dessus. Une évaluation plus à jour est obtenue, incluant toutes les métriques supervisées identifiées lors de la rédaction de notre article (Zaidi, Boilard, Gagnon, & Carbonneau, 2020). Au lieu d'évaluer la capacité des métriques à définir un meilleur modèle d'apprentissage, la capacité des métriques de définir des représentations comme meilleures est évaluée. De considérables désaccords sont à nouveau observés. Finalement, nous profitons du contexte de cette évaluation afin de discuter des difficultés à déterminer la véracité des mesures lorsque les représentations évaluées sont obtenues par l'entraînement de modèles non-supervisés.

2.1 Méthodologie

L'objectif de cette expérience est de reproduire les corrélations obtenues dans (Locatello, et al., 2019) avec quelques différences clés. Tout d'abord, nous utilisons un nombre considérablement plus élevé de métriques. Les métriques utilisées correspondent à celles identifiées en date de la publication de notre papier (Zaidi, Boilard, Gagnon, & Carbonneau, 2020). Il s'agit de *Z-diff* (Higgins, et al., 2017), *Z-min Variance* (Kim & Mnih, 2018), *Z-max Variance* (Kim, Wang, Sahu, & Pavlovic, 2019), *IRS* (Suter, Miladinović, Schölkopf, & Bauer, 2019), *DCI Lasso* / *DCI RF* (Random Forest) (Eastwood & Williams, 2018), *SAP* (Kumar, Sattigeri, & Balakrishnan, 2018), *Modularity* / *Explicitness Score*, MIG (Chen, Li, Grosse, & Duvenaud, 2018), MIG-sup (Li, Murkute, Gyawali, & Wang, 2020), JEMMIG

(Do & Tran, 2020) et DCIMIG (Sepliarskaia, Kiseleva, & de Rijke, 2020). Ces métriques ont été implémentées et distribuées publiquement[2].

L'évaluation de (Locatello, et al., 2019) utilise une multitude de modèles non-supervisés afin d'obtenir les représentations à évaluer. Notre méthodologie n'utilise qu'un seul modèle, soit l'auto-encodeur avec régularisation β-VAE (Higgins, et al., 2017). L'utilisation d'un seul modèle évite d'obtenir un biais de corrélation positive lorsque tous les métriques sont en accord sur la performance représentative globale des modèles.

Les modèles sont entraînés sur deux ensembles de données distincts, soit *Cars3d* (Reed, Zhang, Zhang, & Lee, 2015) et *SmallNORB* (LeCun, Huang, & Bottou, 2004). Dans ces ensembles, les facteurs génératifs et leurs valeurs possibles sont préétablis et une observation est générée artificiellement pour chacune des combinaisons possibles de facteurs génératifs. Cela forme un ensemble avec des facteurs à population balancée et dont la qualité descriptive des étiquettes est complète. Nous référons à ce type d'ensemble de données comme des « *ensembles jouets* ». L'utilisation d'ensembles jouets permet d'évaluer les métriques dans un contexte favorable où seulement les facteurs évalués existent dans les observations de l'ensemble. Ces ensembles favorisent l'obtention de représentations où les dimensions de codes concordent aux facteurs génératifs d'un ensemble de données de façon statistiquement indépendante, ce que l'on dénomme comme une représentation factorielle (Ridgeway, 2016).

2.2 Présentation des résultats

Tout d'abord, des modèles β-VAE sont entraînés et configurés avec chacune des combinaisons des hyperparamètres suivants : la puissance de régularisation $\beta \in \{0.001, 0.01, 0.1, 1, 10, 100\}$ et la dimensionnalité de l'espace de représentation $d \in \{2, 4, 8, 16, 32, 64\}$. Les autres hyperparamètres sont sélectionnés afin de reproduire les résultats obtenus dans (Locatello, et al., 2019). Les entraînements sont effectués sur 300k

[2] https://github.com/ubisoft/ubisoft-laforge-DisentanglementMetrics

étapes avec un optimisateur *Adam* ainsi qu'une taille de lot de 64. Le démêlage des représentations obtenues suite à l'entraînement est ensuite mesuré avec l'ensemble des métriques identifiées. Pour chacune des métriques, le rang des configurations d'hyperparamètres $\{\beta, d\}$ est défini selon l'ordre décroissant des mesures obtenues. La *Kendall Rank Correlation* est utilisée pour définir la corrélation entre les métriques à l'aide de leurs rangs respectifs obtenus. Les corrélations distinctes des métriques sur les ensembles *Cars3d* et *SmallNORB* sont respectivement présentés à la Figure 2.1 et la Figure 2.2.

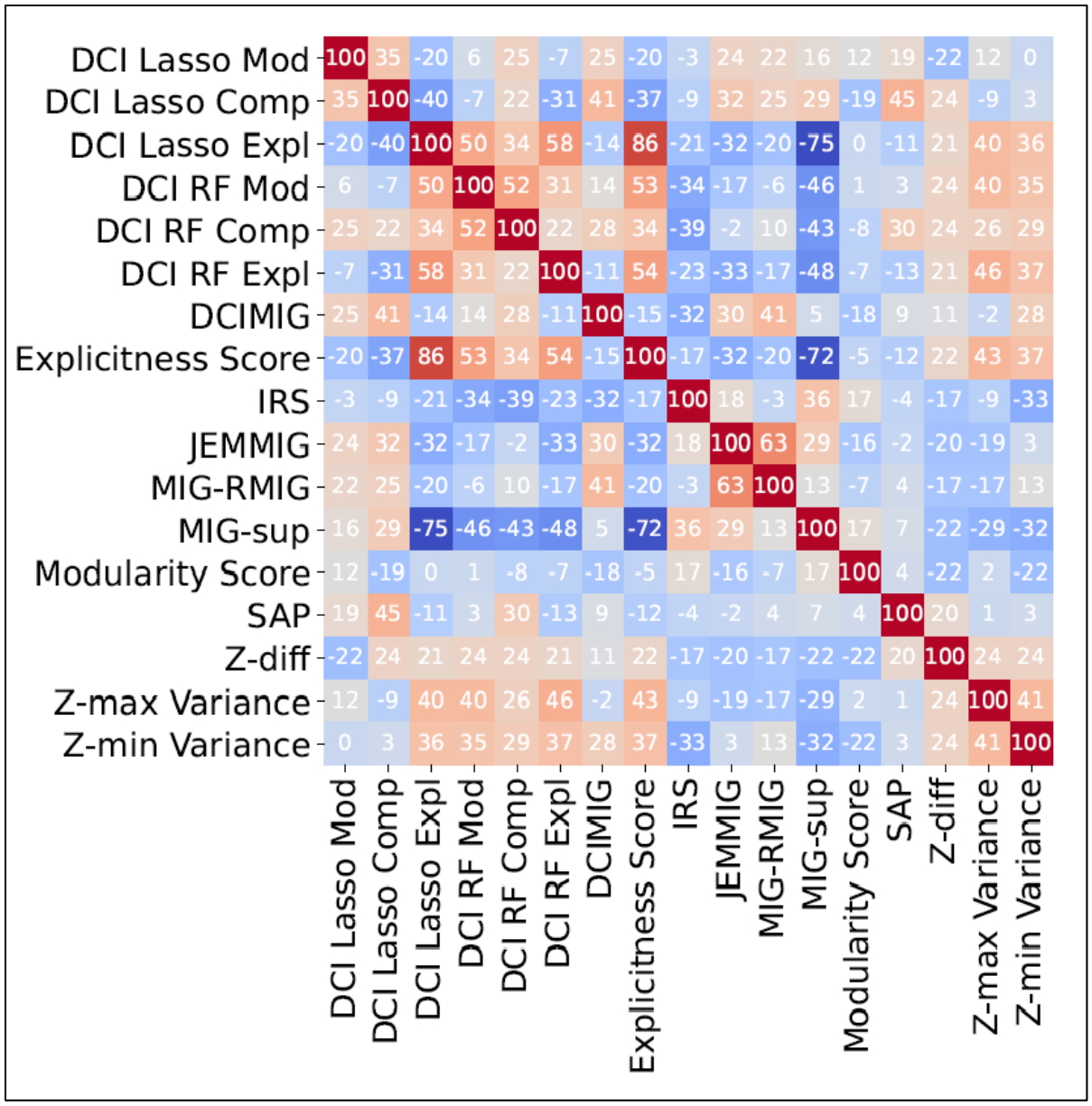

Figure 2.1 Corrélation Kendall (x100) des rangs de configuration d'hyperparamètres obtenus entre toutes les métriques sur l'ensemble de données Cars3d
tiré de Zaidi, Boilard, Gagnon, & Carbonneau (2020, p.9)

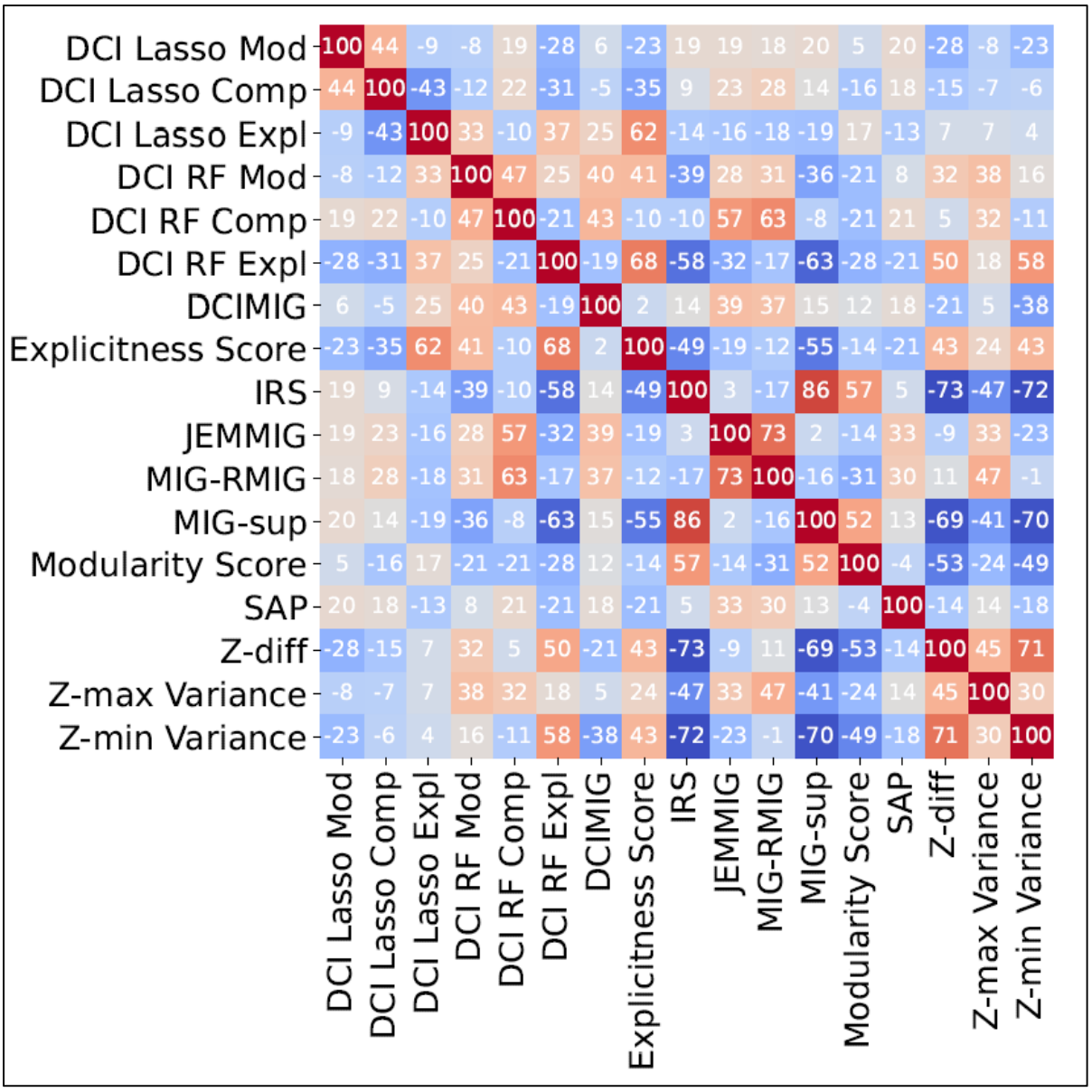

Figure 2.2 Corrélation Kendall (x100) des rangs de configuration d'hyperparamètres obtenus entre toutes les métriques sur l'ensemble de données SmallNORB
tiré de Zaidi, Boilard, Gagnon, & Carbonneau (2020, p.9)

2.3 Interprétation des résultats

L'ensemble des corrélations obtenues sont plutôt faibles, à quelques exceptions près pour certaines métriques avec de fortes similarités, tels que *Z-diff / Z-min Variance* et *MIG / JEMMIG*. Les faibles corrélations, et même la présence de corrélations négatives, démontrent que différentes métriques favorisent différentes propriétés représentatives. Même les métriques censées mesurer la même propriété du démêlage ne sont généralement pas fortement corrélées les unes avec les autres. Ce manque de concordance entre les métriques est aussi observable dans d'autres publications (Locatello, et al., 2019) (Abdi, Abolmaesumi, & Fels, 2019).

2.4 Discussion

Les résultats obtenus démontrent que les métriques ne concordent pas les unes avec les autres dans la section précédente. Comme présenté, les résultats démontrent qu'il est difficile de quantifier le démêlage dans un contexte pratique où l'on cherche à déterminer une meilleure représentation parmi plusieurs. De plus, les difficultés se manifestent malgré l'utilisation d'ensembles jouets supposés favoriser l'obtention de représentations factorielles.

Toutes les études présentant des résultats de corrélation entre les métriques évitent d'interpréter les corrélations obtenues et se limitent à seulement les présenter (Locatello, et al., 2019) (Sepliarskaia, Kiseleva, & de Rijke, 2020). Nous faisons de même dans cette expérience, puisqu'il serait risqué d'interpréter tout résultat obtenu avec les métriques lorsque les relations codes/facteur dans la représentation sont inconnues. Par exemple, cette limitation empêche d'identifier si une représentation contient des dimensions en relations linéaires ou non-linéaires avec un facteur ou si de l'information relative à des facteurs sans-intérêt s'est introduite dans les codes. Rendre accessibles ces informations relatives aux codes/facteurs dans une représentation donnée simplifierait beaucoup la vérification et la caractérisation de l'influence de différentes propriétés représentatives sur les mesures obtenues. Nous motivons le prochain chapitre par cette limitation, où nous proposons de

caractériser les métriques sur des représentations où les propriétés représentatives sont connues.

CHAPITRE 3

SCÉNARIOS REPRÉSENTATIFS POUR L'ÉVALUATION DES MÉTRIQUES

Dans ce chapitre, nous proposons un banc de test fait sur mesure pour la caractérisation de métriques de démêlage. D'abord, nous définissons des propriétés désirables pour toutes métriques afin de mieux supporter les expériences proposées. Celles-ci soulignent les différences explicites entre les mécanismes de différentes métriques. Ensuite, une évaluation des métriques sur des représentations générées artificiellement est proposée. Dans ces représentations artificielles, les relations entre les dimensions de codes et les facteurs sont préétablies. Ces relations visent des qualités spécifiques généralement contrôlables afin de faciliter la caractérisation des métriques.

À partir des résultats obtenus, plusieurs nouveaux cas d'erreurs sont identifiés. Le banc de test élaboré pour évaluer les métriques ainsi que les résultats obtenus sur ceux-ci font partie de notre article (Zaidi, Boilard, Gagnon, & Carbonneau, 2020) présentement en évaluation pour publication dans « *IEEE Transactions on Pattern Analysis and Machine Intelligence* ». Au meilleur de nos connaissances, il s'agit de la première étude en profondeur des métriques visant leur caractérisation sur des scénarios représentatifs prédéfinis et contrôlables.

3.1 Travaux en relation avec notre étude

Dans l'ensemble de la littérature, peu d'études mesurent des représentations avec l'ensemble des métriques de démêlage disponibles au moment de leur rédaction. Il y a donc peu de littérature discutant de l'effet de différentes qualités représentatives sur les mesures, et qui tentent d'interpréter la différence entre les mesures obtenues.

Dans (Abdi, Abolmaesumi, & Fels, 2019), des modèles d'apprentissage de représentation sont classés au lieu des représentations elles-mêmes. Les résultats démontrent que la définition du modèle le plus performant varie d'une métrique à l'autre. De plus, le modèle

défini comme globalement mieux noté sur l'ensemble de métrique ne correspond pas au modèle que les auteurs auraient sélectionné suite à une observation subjective.

Dans (Locatello, et al., 2019), l'étude se limite à calculer la corrélation entre les métriques sur des représentations obtenues à partir de divers ensembles de données et diverses configurations de modèles non-supervisés. Cette étude n'explicite pas pourquoi certaines métriques représentant les mêmes caractéristiques sont non corrélées, ni pourquoi certaines métriques représentant différentes caractéristiques sont corrélées et ni la dégradation de la corrélation lorsque des représentations provenant d'ensemble de données davantage complexes sont mesurées. Les potentielles différences entre les représentations obtenues sont au préalable inconnues, ce qui complique l'interprétation des résultats.

Sinon, (Sepliarskaia, Kiseleva, & de Rijke, 2020) est à notre connaissance la seule étude cherchant à caractériser les métriques. La caractérisation est limitée à vérifier si les métriques sont bien calibrées relativement aux qualités censées être mesurées. Cette calibration se résume à **1)** si elles sont capables de retourner une mesure maximale seulement lorsque les qualités supposées être mesurées par la métrique sont présentes dans la représentation, ainsi que **2)** la capacité à retourner une mesure minimale seulement lorsque ces mêmes qualités y sont absentes. Il est conclu que la majorité des métriques ne respectent pas ces deux requis.

Notre banc de test est en quelque sorte une bonification de l'étude menée par (Sepliarskaia, Kiseleva, & de Rijke, 2020). En plus de la propriété de calibration des métriques vérifiée dans ce dernier ouvrage, nous définissons davantage de propriétés de métriques idéales à respecter. Les scénarios que nous définissons permettent de caractériser les métriques selon ces propriétés de métriques définies en plus de cibler des propriétés quantifiables du démêlage.

3.2 Propriétés de métriques

Dans la section 1.1, différentes propriétés mesurables du démêlage ont été présentées. Dans cette section, nous définissons plutôt de nouvelles propriétés de métriques que nous définissions désirables dans le contexte de nos expériences.

3.2.1 Cohérence des mesures

Tout d'abord, il est important de savoir ce qu'une métrique évalue exactement afin de pouvoir bien interpréter les mesures obtenues. En d'autres mots, une métrique doit pouvoir précisément mesurer une ou plusieurs propriétés identifiables du démêlage. Cette capacité d'interpréter est seulement possible lorsque la cohérence des mesures a été établie. Ainsi, il est important d'éviter les cas d'erreurs similaires à ceux identifiés dans (Kim & Mnih, 2018) (Sepliarskaia, Kiseleva, & de Rijke, 2020). Les métriques mesurant des propriétés uniques ont déjà été identifiées dans la Figure 1.2.

La plage des résultats des métriques devrait être *calibrée* entre les mêmes valeurs afin de faciliter la comparaison des mesures provenant de diverses métriques. Nous définissons la mesure de « 1 » comme maximale, seulement obtenue par la mesure d'une représentation parfaite, et « 0 » comme minimale, seulement obtenue par la mesure d'une représentation aléatoire et indépendante des facteurs. Les métriques présentées dans le CHAPITRE 1 n'étant pas originalement ainsi calibrées dans leurs publications respectives ont été normalisées afin de rapporter leur mesure entre les bornes [0,1]. La section 3.3.1 démontre que la normalisation n'est pas toujours suffisante pour obtenir des mesures calibrées. L'évaluation de la calibration est aussi effectuée dans (Sepliarskaia, Kiseleva, & de Rijke, 2020), mais nous l'effectuons sur une plus grande quantité de métriques et de façon empirique sur l'ensemble des scénarios proposés.

Pour que les mesures d'une caractéristique du démêlage soient cohérentes, elles doivent aussi refléter adéquatement une variation de la qualité de cette caractéristique. Ainsi, les métriques devraient retourner des mesures entre des bornes calibrées évoluant de façon strictement monotone selon la qualité des propriétés représentatives ciblées. Lorsqu'une métrique est censée mesurer une propriété du démêlage quelconque, il est indésirable que des mesures ne respectant des degrés différents de propriétés ciblées soient obtenus, puisque cela complique l'interprétation du rang de qualité des représentations. Nous évaluons l'évolution des mesures par des changements à bonds égaux de la qualité explicite dans la section 3.3.1, et de la modularité et compacité dans la section 3.3.2.

3.2.2 Absence d'assomptions relationnelles

Certaines applications favorisent des relations simples, voire linéaires, comme représentation de facteurs. (Ridgeway & Mozer, 2018) (Kumar, Sattigeri, & Balakrishnan, 2018). Afin de se spécialiser à ce cas, une métrique pourrait assumer une relation monotone entre les codes et facteurs. Cette monotonie permet de naviguer intuitivement dans l'espace des données par une augmentation graduelle de la valeur des dimensions de codes associées à un facteur continu. Cependant, certains facteurs ne peuvent être décrits avec cette assomption d'augmentation graduelle. Par exemple, une telle navigation est insensée lorsque l'on évalue le démêlage de facteurs catégoriques. Cela s'explique par le fait que sa distribution dans une représentation est multimodale, soit différentes catégories sont manifestées dans différents modes de l'ensemble des codes (Guo, Wang, & Wang, 2019).

Un exemple de facteur continu incompatible avec plusieurs assomptions relationnelles est la rotation. Tel que défini dans (Ridgeway & Mozer, 2018), ce facteur est mieux représenté par une relation non-linéaire à multiples variables non-monotones. Dans certaines études, le démêlage de la rotation a parfois été observé par traverse latente à une seule variable. Dans ce cas spécifique, l'assomption univariée complique les impressions qualitatives des

évaluateurs. (Chen, Li, Grosse, & Duvenaud, 2018) (Higgins, et al., 2017) (Do & Tran, 2020) (Kim, Wang, Sahu, & Pavlovic, 2019).

De telles complications sont évitées si le démêlage est évalué avec une méthode ne contenant aucun mécanisme assumant la nature de la relation entre les codes et facteurs. Dans la section 3.3, l'ensemble des scénarios représentatifs proposés exposent divers cas incompatibles avec les assomptions d'une métrique. Dans la section 3.3.3, les assomptions linéaires des métriques sont caractérisées sur des représentations monotones où la non-linéarité est graduellement augmentée. Dans la section 3.3.4, la capacité des métriques à capturer une représentation non-linéaire, non-monotones et à multiples variables est vérifiée.

3.2.3 Robustesse des mesures de modularité et compacité à l'induction de bruit

Tout d'abord, supposons un cas où une représentation modulaire, compacte, mais modérément explicite d'un facteur identifié est mesurée. Une mesure de qualité explicite modérée n'implique pas que les mesures de modularité ou compacité soient nécessairement négativement affectées. Par exemple, de tels niveaux de modularité, compacité et qualité explicite peuvent démontrer qu'un facteur soit seulement partiellement distillé dans la représentation, et/ou la possibilité d'une intrusion de facteurs non-identifiés dans les dimensions de codes. Il est attendu que les mécanismes internes de métriques soient capables de discerner de tels cas. Les mesures de modularité/compacité devraient donc être robustes à une intrusion de bruit émulant ces cas, et cela est évalué dans la section 3.3.1.

3.2.4 Robustesse des mesures aux dimensions impertinentes de codes

Supposons un autre cas où une représentation parfaite des facteurs d'intérêt est apprise, mais où quelques dimensions de codes supplémentaires représentent des facteurs non-identifiés. Il est nécessaire que les mesures de qualités représentatives ne soient pas biaisées par ce cas. Par exemple, il ne serait pas idéal qu'une mesure de qualité explicite pénalise une telle

représentation qui contiendrait toute l'information nécessaire à la prédiction d'un facteur. De façon semblable, il serait inapproprié qu'une telle dimension de code soit considérée dans la mesure globale de la modularité et la compacité. Certaines métriques ont des mécanismes afin de rejeter des dimensions de codes inactives ou impertinentes, rendant possible de correctement juger la qualité d'une représentation comme bonne malgré leur présence (Eastwood & Williams, 2018) (Kim & Mnih, 2018). Ainsi, il est constaté que la présence de ce mécanisme constitue en une propriété idéale de métrique. La robustesse des mesures aux dimensions impertinentes de codes est évaluée à la section 3.3.5 en omettant graduellement de plus en plus de facteurs connus d'une mesure à l'autre.

3.2.5 Stabilité paramétrique des mesures

Si une métrique contient un hyperparamètre interne quelconque, il est idéal qu'une configuration générique existe où la performance est bonne dans tous les cas. Dans le cas contraire, il faut procéder à une optimisation des hyperparamètres internes pour chacune des représentations évaluées. Si ceux-ci sont mal optimisés, cela pourrait causer des problèmes de stabilité, tel que le sousapprentissage et le surapprentissage. Des exemples de tels hyperparamètres sont des paramètres d'entrainement de modèles internes d'une métrique (e.g le poids de régularisation d'un modèle prédictif), la granularité de discrétisation des facteurs et codes, le batch size, etc. La meilleure façon de mitiger les problèmes relatifs aux hyperparamètres est bien évidemment d'en minimiser le nombre. Plusieurs métriques démontrent des cas d'instabilité paramétrique dans les résultats de la section 3.3.

3.3 Scénarios d'évaluation

Cette section cherche à caractériser les métriques selon leurs potentielles propriétés décrites à la section 3.2. Nous proposons des scénarios où des relations entre chacune des dimensions de codes et les facteurs ciblent des qualités représentatives précises. Contrairement aux évaluations effectuées dans (Locatello, et al., 2019) (Abdi, Abolmaesumi, & Fels, 2019)

l'approche proposée ici dégage les qualités représentatives inconnues d'une représentation compliquant l'évaluation des métriques.

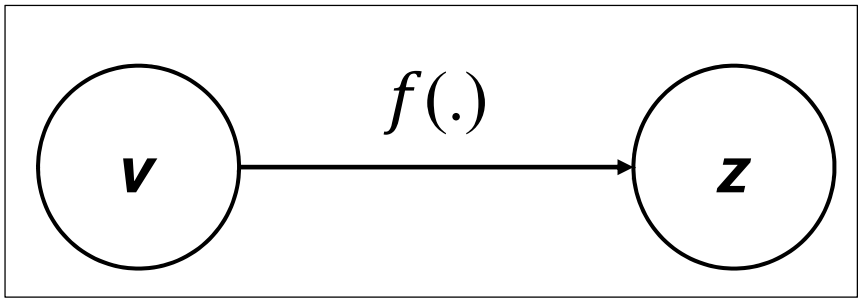

Figure 3.1 Nomenclature ajustée aux scénarios d'évaluation

Tel qu'illustré à la Figure 3.1, nous proposons à des fins d'évaluation qu'un code $\boldsymbol{z} = \{z_1, z_2, \dots, z_d\}$ soit directement dicté par des facteurs $\boldsymbol{v} = \{v_1, v_2, \dots, v_M\}$ par une fonction $f(.)$. Cette fonction est unique à chacun des scénarios proposés et contient potentiellement un paramètre α dictant la qualité des propriétés ciblées dans les codes générés. Ainsi, la fonction de représentation de la Figure 3.1 devient $f(.) \rightarrow f(\boldsymbol{v}|\alpha)$. Par exemple, dans le scénario de la section 3.3.1, α contrôle la présence de bruit uniforme dans l'ensemble des dimensions de $\boldsymbol{z}$. Quant au scénario 3.3.4, le contrôle d'une qualité représentative n'est pas nécessaire et la fonction de représentation devient $f(.) \rightarrow f(\boldsymbol{v})$. Dans l'ensemble des scénarios, les facteurs génératifs individuels $v_i \in \boldsymbol{v}$ sont échantillonnés selon des distributions uniformes. Lorsqu'une métrique le nécessite, les facteurs sont discrétisés en 10 bornes également distancées, résultant en 10^M combinaisons possibles de réalisations discrètes des facteurs.

Suivant la nomenclature de la section 1.2, les métriques reçoivent des ensembles $V^{(N)}$ et $Z^{(N)}$ de population $N = 20k$. La dimensionnalité des codes d de $\boldsymbol{z} = \{z_1, z_2, \dots, z_d\}$ dépend de la fonction $f(.)$ définie par les scénarios. Tous les résultats présentés sont une moyenne des mesures obtenues en définissant 100 « *random seeds* » distincts. Ceux-ci influencent l'échantillonnage de l'ensemble $V^{(N)}$ ainsi que les mécanismes internes des métriques faisant usage de générateurs de nombres aléatoires (e.g. Entraînement d'un régresseur ou sélection

de sous-ensembles pour la famille de métriques interventionnels). Il a été observé sur l'ensemble des scénarios que les conclusions sur les résultats obtenus sont similaires pour tout nombre de facteurs génératifs $M \geq 2$.

3.3.1 Induction de bruit dans une représentation parfaite

Dans le scénario suivant, la robustesse au bruit des mesures de modularité et compacité est caractérisée selon la propriété définie à la section 3.2.3. De plus, la formulation de la relation code/facteur permet aussi la vérification de la calibration des métriques, un requis pour leur cohérence (section 3.2.1). La fonction utilisée afin de générer les codes est la suivante :

$$\boldsymbol{z} = f(\boldsymbol{v}|\alpha) = (1-\alpha)\boldsymbol{v} + \alpha\boldsymbol{n} \tag{3.1}$$

Où intuitivement, les codes $\boldsymbol{z}$ sont des copies des facteurs $\boldsymbol{v}$**,** mais graduellement remplacés par du bruit par l'augmentation du paramètre $\alpha \in [0,1]$. Ce bruit est échantillonné selon une distribution uniforme $\boldsymbol{n} \sim U(0,1)$. Les facteurs $\boldsymbol{v} = \{v_1, v_2, \dots, v_M\}$ et codes $\boldsymbol{z} = \{z_1, z_2, \dots, z_d\}$ sont de dimensionnalité $M = d = 8$. Les facteurs sont uniformément échantillonnés selon $v_i \sim U(0,1)$. L'ensemble $V^{(N)}$ est échantillonné et les codes associés aux facteurs échantillonnés $\boldsymbol{z} = f(\boldsymbol{v}|\alpha)$ sont calculés afin de construire l'ensemble $Z^{(N)}$. La Figure 3.2 illustre les résultats moyens pour chacune des métriques sur le scénario d'induction de bruit.

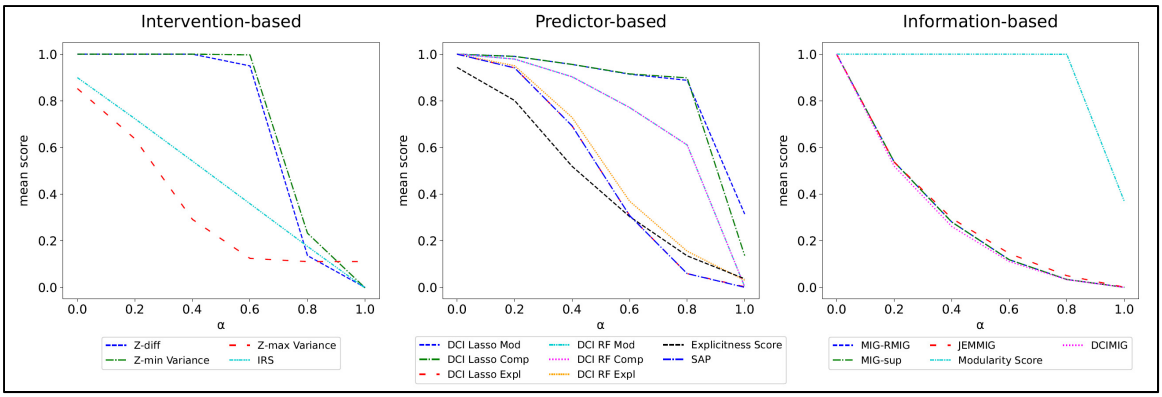

Figure 3.2 Résultats moyens obtenus pour le scénario d'induction de bruit dans une représentation parfaite
tiré de Zaidi, Boilard, Gagnon, & Carbonneau (2020, p.10)

Dans ce scénario, le comportement idéal à l'augmentation linéaire de α est une dégradation relativement linéaire de la qualité explicite, et une certaine robustesse des mesures de modularité et compacité pour $\alpha \leq 0.8$. Les courbes de la Figure 3.2 devraient idéalement refléter ces comportements. La Figure 3.2 démontre une dégradation attendue et appropriée des mesures de qualité explicite relativement à la réduction de α. Pour DCI *Modularity/Compactness*, il est inévitable ces mesures soient affectés par une dégradation de la capacité prédictive avec l'introduction du bruit. Tout de même, elles démontrent une robustesse appropriée, avec le pire cas étant *DCI RF Modularity / Compactness*, qui mesurent les deux leur propriété respective à $\sim$0.8 pour $\alpha = 0.6$ (60% de bruit) et $\sim$0.65 en moyenne pour $\alpha = 0.8$ (80% de bruit). Quant à la famille *information-based*, les métriques de modularité et compacité ne démontrent pas cette même robustesse. Cela est dû au fait que le calcul de l'information mutuelle à l'intérieur des métriques cause les mesures à dépendre davantage de la qualité explicite de la représentation.

Quant à eux, les métriques *Z-diff* et *Z-min variance* démontrent une forte robustesse pour $\alpha \leq 0.6$. Cependant, *Z-Max Variance* ne partage pas la même robustesse. Pour cette dernière métrique, la variance des codes est mesurée en sélectionnant un ensemble $\mathcal{V}^{(\mathcal{N})} \subset V^{(N)}$ où la valeur de chacun des facteurs v_i du sous-ensemble $\mathcal{V}^{(\mathcal{N})} \ni \boldsymbol{v} \ni v_i$ est libre alors que l'on fixe

les $M-1$ autres valeurs de facteurs. Le nombre très limité de combinaisons possibles de facteurs fixables dans l'ensemble $V^{(N)}$ échantillonné avec une population $N = 20k$ et M=8 facteurs combiné au fait que notre implémentation permet le calcul de la variance de $\mathcal{V}^{(\mathcal{N})}$ seulement lorsque sa population est ≥ 2 accentue un cas d'erreur de la métrique, soit la possibilité d'obtenir une plus grande variance dans les dimensions de codes représentant les $M-1$ facteurs fixes plutôt que dans la dimension de code représentant le facteur libre. Cela engendre une confusion dans le classificateur majoritaire et biaise négativement les mesures obtenues.

L'obtention d'une représentation parfaite et aléatoire pour respectivement $\alpha = 0$ et $\alpha = 1$ nous permet de vérifier la bonne calibration des mesures. Les métriques ne retournent pas toutes des mesures calibrées. Premièrement, une bonne calibration exige qu'une mesure parfaite soit retournée seulement lorsqu'une représentation parfaite est mesurée ($\alpha = 0$). Une première exception à cela est la métrique *IRS*. Les valeurs d'un ensemble de facteurs génératifs $\boldsymbol{v} = \{v_1, v_2, \ldots, v_M\}$ sont d'abord échantillonnés uniformément et ensuite discrétisés. La discrétisation des facteurs v_i force tous les éléments d'une même classe d'être homogènes, alors que les dimensions représentatives de codes préservent leur variance. Cela influence les mesures de divergence de la moyenne suivant une intervention sur les facteurs et cause les mesures à être biaisées négativement.

Sinon, l'*Explicitness Score* n'obtient pas une mesure parfaite pour $\alpha = 0$ puisque les auteurs suggèrent l'utilisation d'un classificateur one-vs-rest multi-classe. Ce classificateur est peu adapté à la relation monotonique de ce scénario.

Une métrique bien calibrée nécessite de plus qu'elle retourne une mesure nulle pour les représentations complètement bruitées ($\alpha = 1$). Un premier contrevenant est *Z-Max Variance*. Cette métrique nécessite des codes issus de combinaisons différentes de facteurs. La faible variété de telles combinaisons dans l'ensemble $V^{(N)}$ échantillonné rend inadaptée l'utilisation d'un classificateur majoritaire, ce qui biaise les mesures retournées.

Les métriques DCI Lasso Modularity/Compactness n'arrivent pas non plus à retourner une mesure moyenne nulle, ce qui est causé par la régularisation Lasso du modèle interne entraîné. Celui-ci impose une minimisation de la norme des poids pour la prédiction d'un facteur. Cela cause des mesures instables d'un sous-ensemble $\mathcal{V}^{(N)}$ à l'autre. Cela est dû au fait qu'une dimension de code contient du bruit échantillonné davantage corrélé par hasard avec un facteur, causant un poids à être légèrement plus fort que les autres par hasard. La régularisation privilégie souvent les poids associés à ce hasard et potentiellement au détriment quasi total des autres, ce qui bonifie faussement les mesures individuelles.

Dans le cas de la métrique *Modularity,* la mesure moyenne non-nulle pour $\alpha = 1$ est causée par une normalisation trop sévère permettant difficilement d'obtenir une mesure nulle. Pour obtenir une mesure nulle ou presque nulle, le numérateur doit être encore plus près de la valeur au dénominateur de que d'autres métriques effectuant une normalisation puisque la métrique *Modularity* élève ces deux termes au carré. De plus, cette normalisation rend la métrique beaucoup moins expressive pour toutes représentations générées avec $\alpha \leq 0.8$. Ainsi, la seule plage de variation discernable est pour $\alpha > 0.8$.

3.3.2 Réduction de la modularité et la compacité

Dans ce scénario, des représentations sont générées en effectuant une réduction graduelle combinée de leur modularité et leur compacité sans compromettre leur qualité explicite. Il est attendu que les mesures de compacité et modularité reflètent cette réduction. Nous caractérisons donc la cohérence des mesures de modularité et compacité telle que définie à la section 3.2.1. L'espace de représentation est construit selon $\boldsymbol{z} = f(v) = \boldsymbol{v}R$, où la matrice R effectue une projection des facteurs tel que défini ci bas :

$$R = \begin{bmatrix} 1-\alpha & \alpha & 0 & \cdots & 0 \\ 0 & 1-\alpha & \alpha & \cdots & 0 \\ 0 & 0 & \alpha & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha & 0 & 0 & \cdots & 1-\alpha \end{bmatrix} \tag{3.2}$$

Dans ce scénario, $\alpha \in [0, 0.1, \ldots, 0.5]$. Les éléments $\boldsymbol{z} \in Z^{(N)}$ et $\boldsymbol{v} \in V^{(N)}$ sont configurés comme le scénario précédent (3.3.1). Les dimensions de codes $z_j \in \boldsymbol{z}$ et les facteurs $v_i \in \boldsymbol{v}$ sont complètement modulaires et compactes lorsque $\alpha = 0$. Avec chaque augmentation de α, chacune des dimensions de codes représente avec graduellement plus d'importance un deuxième facteur, et chacun des facteurs est graduellement plus représenté par une deuxième dimension de codes. La Figure 3.3 illustre les mesures moyennes obtenues pour chacune des métriques.

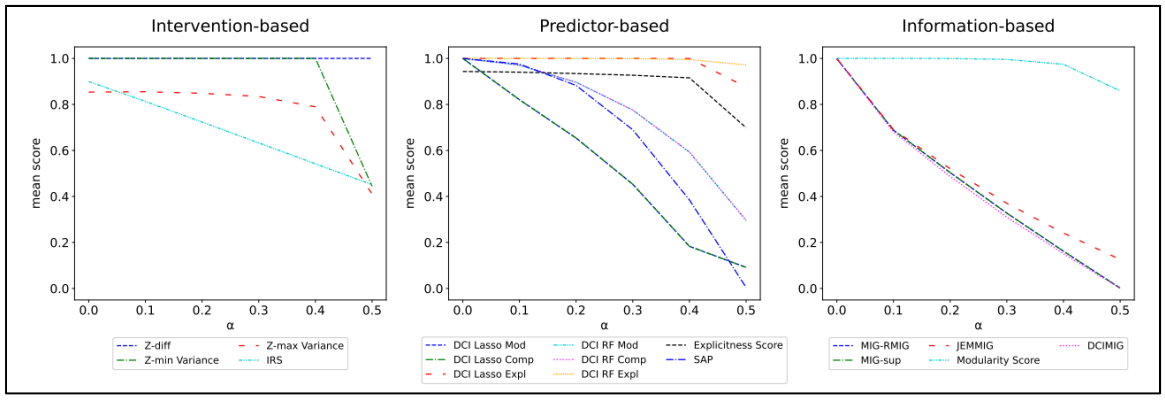

Figure 3.3 Résultats obtenus pour le scénario de réduction de la modularité et la compacité tiré de Zaidi, Boilard, Gagnon, & Carbonneau (2020, p.11)

On observe pour $\alpha > 0$ un comportement pouvant limiter l'utilité des métriques *information-based* ainsi que *SAP*. Ces métriques considèrent que deux facteurs représentant un code ou vice-versa équivaut à une modularité et/ou compacité nulle. Ces métriques contiennent une opération effectuant la soustraction de la meilleure mesure de la relation entre un

code/facteur soustrait par la deuxième meilleure. Dans le cas de ces métriques, cela empêche de quantifier la modularité passé le seuil d'une dimension de code représentative de 2 facteurs et empêche les mesures de compacité de quantifier passé le seuil d'un facteur représenté par 2 dimensions de codes. Ce cas d'erreur est davantage explicité dans le scénario de la section 3.3.4.

Les métriques *Z*-diff, *Z-min variance* et *Z-max variance* sont des mesures décrites comme des mesures de modularité. Cependant, on observe un comportement dans leurs mesures ne reflétant pas l'évolution de la non-modularité du scénario. Dans le cas de *Z-diff*, la mesure retournée est toujours parfaite pour n'importe quel α puisque le classificateur interne arrive tout de même à tracer des seuils de classification pour la prédiction du facteur fixé, peu importe α. Ce cas d'erreur est exactement le même souligné dans (Sepliarskaia, Kiseleva, & de Rijke, 2020), où la constante représentativité de 2 facteurs par une dimension. Dans le cas de *Z-min variance* et *Z-max variance*, la variance des dimensions de codes est mesurée lorsque respectivement un / plusieurs facteurs sont fixés. Ces métriques restent capables de discerner à ~100% la dimension de code avec l'association la plus forte pour $\alpha =< 0.4$. Lorsque $\alpha = 0.5$, une confusion entre les deux dimensions de codes également représentatifs de v_i dans le classificateur majoritaire permet d'obtenir la précision attendue de ~50%.

Sinon, comme explicité à la section 3.3.1, la métrique *Modularity* a une forte tendance à surestimer la représentation, ce qui est encore observable dans ce scénario. La gravité de surestimation nous indique que la métrique n'est pas cohérente avec l'évolution de α.

3.3.3 Relations non-linéaire

La représentativité linéaire des facteurs est fort souvent désirable, mais pas ultimement nécessaire. En fait, un facteur pourrait même être représenté plus explicitement par une relation non-linéaire. Certaines métriques intègrent des mécanismes explicitement adaptés à favoriser les relations linéaires, tel qu'un régresseur linéaire dans DCI Lasso. La motivation

de ce scénario est de découvrir si d'autres métriques contiennent des mécanismes incompatibles aux relations non-linéaires. Dans ce scénario, nous dressons une relation 1:1 entre les codes/facteurs qui est constamment complètement modulaire et complètement compacte. Le scénario introduit un contrôle sur à quel point la relation est non-linéairement explicite. La fonction utilisée afin de générer les codes est la suivante :

$$\boldsymbol{z} = \mathrm{f}(\boldsymbol{v}) = 1000^{-\alpha+0.25} \tan\big(\omega(\mathbf{v} - 0.5)\big) + 0.5 \qquad (3.3)$$

Où $\omega = 2\tan^{-1}\left(\frac{1000^{\alpha-0.25}}{2}\right)$. Tel qu'illustré à la Figure 3.4, la fonction tangente est pratiquement linéaire pour $\alpha = 0$, et graduellement plus non-linéaire par l'augmentation graduelle de α, où $\alpha \in [0, 0.2, \ldots, 1]$.

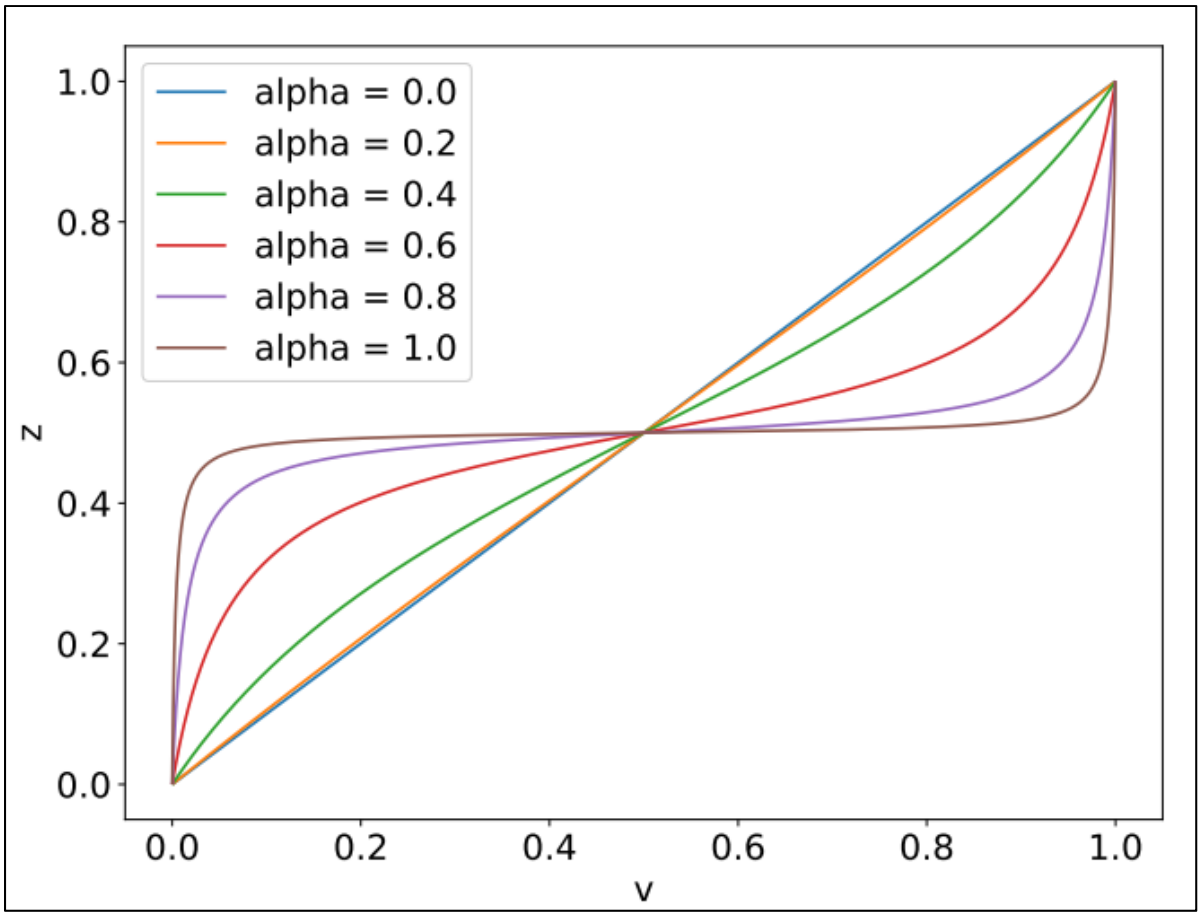

Figure 3.4 Variation de la relation entre ***z*** et ***v*** selon l'évolution de α, pour le scénario non-linéaire
tiré de Zaidi, Boilard, Gagnon, & Carbonneau (2020, p.13)

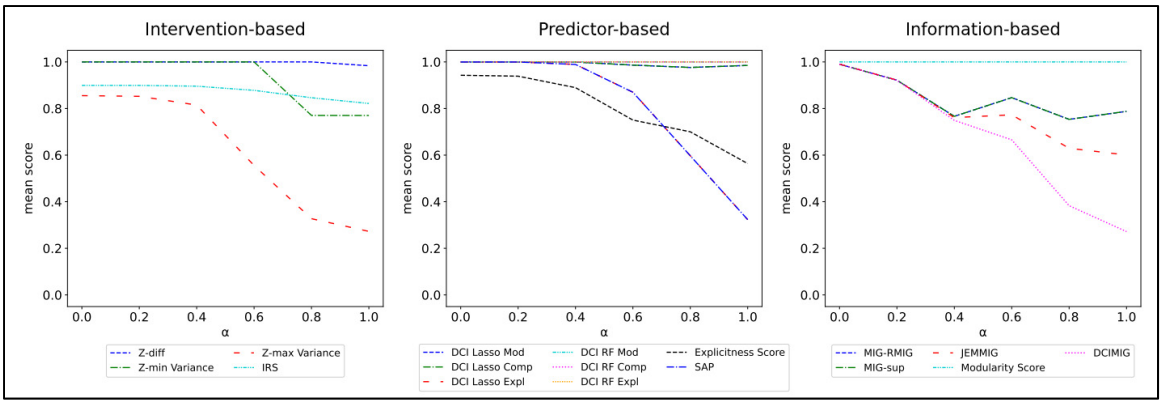

Figure 3.5 Résultats obtenus pour le scénario non-linéaire
tiré de Zaidi, Boilard, Gagnon, & Carbonneau (2020, p.13)

La Figure 3.5 illustre les résultats moyens obtenus pour ce scénario. On y observe que les métriques *intervention-based* sont affectées par cette non-linéarité à des degrés différents. Suite à la discrétisation des facteurs, la fonction de répartition des facteurs $v_i \in \boldsymbol{v}$ est uniforme. Lorsque l'on applique la relation non-linéaire de l'équation 3.3, différentes fonctions de répartition des codes sont obtenues dépendant de α, tel qu'illustré à la Figure 3.6. Pour $\alpha = 0$, la fonction de répartition des codes est virtuellement uniforme. Cependant, avec l'augmentation de α, la fonction de répartition prend la forme d'une distribution normale. Ainsi, l'augmentation de α rend plus rares les valeurs minimales et maximales des dimensions de codes, ce qui a le potentiel de fortement influencer les estimations statistiques locales du sous-ensemble des dimensions de codes (variance, moyenne, divergence maximale, etc). Ainsi, par la discrétisation en ensembles aux bornes également distancées, ces métriques effectuent une assomption de distribution suffisamment uniforme des dimensions de codes de $\mathcal{Z}^{(\mathcal{N})}$ évalués rendant ces métriques incompatibles avec les représentations non-linéairement explicites. Cela pourrait être réglé en augmentant l'hyperparamètre de la population $\mathcal{N}$ des sous-ensembles interventionnels. Cependant, cela n'est pas nécessairement possible en fonction de l'ensemble de données utilisé, et le simple fait de suggérer cette solution démontre un problème d'instabilité paramétrique.

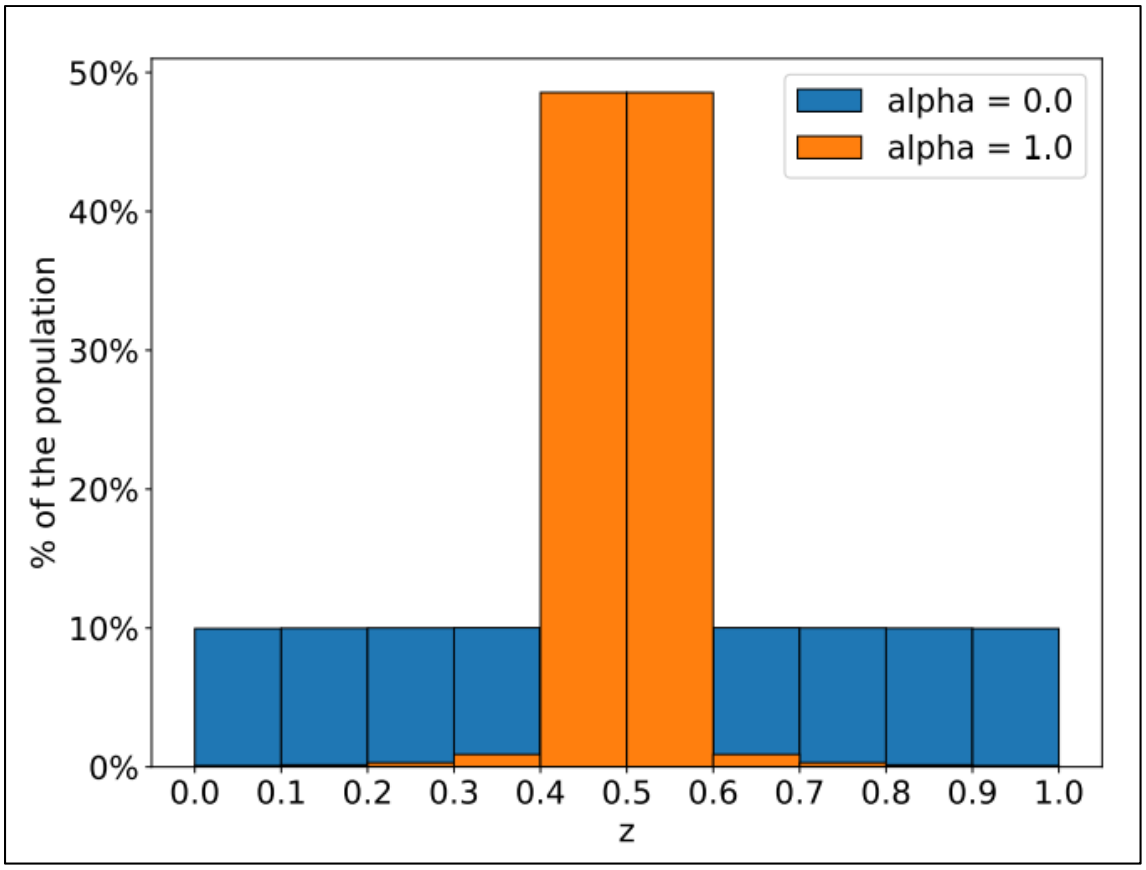

Figure 3.6 Illustration de la discrétisation de la fonction $f(v_i)$ du scénario non-linéaire tiré de Zaidi, Boilard, Gagnon, & Carbonneau (2020, p.13)

Tel qu'attendu pour la famille *predictor-based*, les métriques d'*explicitness* faisant usage d'un régresseur non-linéaire performent mieux que leurs contreparties linéaires. Il est important de discerner les résultats de *DCI Lasso Modularity* et *DCI Lasso Compactness*, qui eux restent relativement élevés malgré l'utilisation de régresseurs linéaires. Ceux-ci restent capables d'optimiser les poids connectés aux dimensions de codes explicites au facteur à prédire puisque les régresseurs linéaires sont capables d'être optimisés sur les fonctions monotoniques. Nous considérons tout de même que DCI Lasso effectue une assomption de relation linéaire incompatible avec les relations non-linéaires puisque les mesures de *Modularity*/*Compactness* sont tout de même affectées par α, de même pour *DCI Lasso Explicitness*.

SAP effectue aussi de fortes assomptions linéaires puisque les coefficients de détermination R^2 sont des mesures de la prédictibilité linéaire d'une fonction, et sont utilisés pour quantifier la relation entre chacune des dimensions de codes et les facteurs. SAP fournit donc des mesures moins élevées lors de la présence de non-linéarité.

Les métriques *information-based* nécessitent non-seulement la discrétisation des facteurs, mais aussi des codes. La fonction non-linéaire de l'équation 3.3 est discrétisée en ensembles égaux sur la valeur des codes afin d'illustrer la fonction de répartition des codes à la Figure 3.6. Cela résulte en une population inégale concentrée au centre lorsque $\alpha > 0$. Cette inégalité issue de la discrétisation est problématique pour les métriques *information-based*. Pour chacune des métriques, l'information mutuelle est calculée avec les densités de probabilités jointes $P(v_i, z_j)$. Lorsque des codes à relations non-linéaires sont discrétisés en ensembles égaux, une classe de dimension de code z_j contient des membres qui sont issues de différentes classes d'un facteur v_i. Cela rend impossible d'estimer une forte information mutuelle. De plus, ces métriques sont normalisées selon l'entropie de la distribution uniforme des réalisations discrètes $H(v_i)$. Cette entropie peut être significativement plus élevée que l'information mutuelle maximale de $I(v_i, z_j)$ puisque $I(v_i, z_j) \leq \min\left(H(v_i),\ H(z_j)\right)$. Ainsi, plus grand est le paramètre α, plus petite est l'entropie $H(z_j)$ et intrinsèquement l'information mutuelle maximale de $I(v_i, z_j)$, et donc plus fortement les estimations du démêlage sont négativement affectées. Les métriques *information-based* assument donc que l'entropie de la distribution des codes est la même que l'entropie de la distribution des facteurs. Cette assomption rend les métriques d'information mutuelle incompatibles avec les relations non-linéaires.

3.3.4 Modulaire, mais non-compacte

Tel qu'expliqué dans la section 1.1, il peut être impossible d'obtenir une représentation compacte sans compromettre la qualité explicite pour certains facteurs. Nous voulons nous assurer que les mesures de modularité et de qualité explicite ne sont pas affectées par des variations exclusives de la compacité. Dans le scénario suivant, $\boldsymbol{v}$ est défini comme un ensemble de M=4 facteurs. Les métriques sont évaluées sur 3 sous-scénarios de représentativité différente de ces facteurs.

Un premier sous-scénario émule une représentation trigonométrique d'angles, une représentation dont la compacité ne compromet pas la qualité explicite de la représentation. Les codes obtenus sont donc équivalents à $\boldsymbol{z} = [\sin v_1 , \cos v_1 , \dots , \sin v_M , \cos v_M]$, où $v_i \sim U(0,2\pi)$. Le deuxième sous-scénario est une simple transposition doublée de facteurs $f(v_i) = [v_i, v_i]$, où $v_i \sim U(0,1)$, où $\boldsymbol{z} = [v_1, v_1, \dots , v_M, v_M]$. Une dernière représentation est générée afin de mieux caractériser l'effet du nombre de codes représentatifs relativement à la dernière représentation, où une transposition quadruple est utilisée, soit $\boldsymbol{z} = [v_1, v_1, v_1, v_1, \dots v_M, v_M, v_M, v_M]$.

Les trois sous-scénarios visent la caractérisation de ces deux propriétés :

1. Vérifier l'absence d'assomptions qui seraient incompatibles avec la relation trigonométrique du sous-scénario 1. Nous vérifions cela en nous assurant que les mêmes mesures pour l'ensemble des propriétés soient obtenues entre les sous-scénarios 1 et 2. Les résultats obtenus relativement à cet objectif supporteront ceux du scénario non-linéaire de la section 3.3.3 en utilisant plutôt une relation non-linéaire à multiples variables.

2. Quantifier la cohérence des mesures en fonction de différents niveaux de compacité. Dans l'ensemble des sous-scénarios, les mesures de *modularité* doivent être les mêmes.

De plus, les mesures de compacité doivent être < 1 et diminuer entre le scénario 2 et 3 et afin d'être cohérentes.

Le Tableau 3.1 présente les résultats de métriques obtenus pour chacun des sous-scénarios décrits.

Tableau 3.1 Relations et résultats du scénario « Modulaire, mais non-compacte » tiré de Zaidi, Boilard, Gagnon, & Carbonneau (2020, p.12)

	Intervention-based				***Predictor-based***								***Intervention-based***				
	Z-diff	**Z-min Variance**	**Z-max Variance**	**IRS**	**DCI Lasso Modularity**	**DCI Lasso Compactness**	**DCI Lasso Explicitness**	**DCI RF Modularity**	**DCI RF Compactness**	**DCI RF Explicitness**	**Explicitness Score**	**SAP**	**MIG**	**MIG-sup**	**JEMMIG**	**Modularity Score**	**DCIMIG**
1) $f(\theta_m) = [\cos\theta_m, \sin\theta_m]$	1.0	1.0	1.0	0.8	0.8	1.0	0.6	1.0	0.7	1.0	1.0	0.6	0.0	0.7	0.4	1.0	0.6
2) $f(\theta_m) = [\theta_m, \theta_m]$	1.0	1.0	1.0	0.9	1.0	1.0	1.0	1.0	0.7	1.0	1.0	0.0	0.0	1.0	0.5	1.0	1.0
3) $f(\theta_m) = [\theta_m, \theta_m, \theta_m, \theta_m]$	1.0	1.0	1.0	0.9	1.0	1.0	1.0	1.0	0.4	1.0	1.0	0.0	0.0	1.0	0.5	1.0	1.0

Voici d'abord les problèmes observables entre les mesures retournées aux sous-scénarios 1) et 2). JEMMIG et DCIMIG sont des mesures holistiques contenant des mécanismes mesurant la compacité. Ces métriques et SAP obtiennent des mesures de compacité différentes entre les sous-scénarios 1) et 2), ce qui corrobore avec les incompatibilités aux relations non-linéaires démontrées à la section 3.3.3. Les mesures de qualité explicites faisant usage de régresseurs linéaires présument évidemment une relation linéaire entre les codes et facteurs, ce qui explique leurs différences.

Dans le cas de mesures de modularité, avec exception IRS due à sa mauvaise calibration déjà identifiée, des inégalités entre les sous-scénarios 1) et 2) sont observables. L'inégalité de *DCI Lasso Modularity* est attribuable à des problèmes d'optimisation du modèle spécifiques au sous-scénario 1). Cette même inégalité des mesures est observable pour les métriques

information-based de modularité*, soit MIG-sup, JEMMIG, Modularity Score* et *DCIMIG*. À noter qu'une analyse plus assidue de mesures moyennes de *Modularity Score* entre le scénario 1) et le scénario 2) démontrent qu'elles sont en réalité différentes d'un ordre de $10e^{-6}$. Ces inégalités sont partiellement attribuables à des assomptions incompatibles aux relations non-linéaires, ce qui a déjà été démontré à la section 3.3.3. Cependant, l'incapacité de ces métriques à définir comme parfaitement modulaire le sous-scénario 1) démontre la présence d'une autre assomption problématique. Il peut être observé que ces métriques sont incapables de définir comme modulaire une dimension des codes explicitant partiellement un facteur unique. Afin d'expliquer l'assomption effectuée, une mesure de modularité pour la famille de métrique *information-based* s'effectue généralement par une soustraction de $I(z_j, v_*)$ par $I(z_j, v_\circ)$, où v_* et $v_\circ$ correspondent à des facteurs représentatifs de z_j respectivement au premier rang et au deuxième rang. Dans le cas du sous-scénario 1), $I(z_j, v_\circ) = 0$ puisque la dimension de code ne représente pas un deuxième facteur. Cependant, il est impossible que la mesure $I(z_j, v_*)$ soit maximale pour ce scénario. Cela est parce que v_* peut seulement être complètement décrit par une combinaison de 2 dimensions de codes dans le sous-scénario 1). Les mesures d'information mutuelle tels qu'implémentés dans la famille *information-based* font donc une assomption de relation univariée entre les codes et facteurs. La présence de mécanismes similaires dans SAP amène aussi à conclure que celui-ci contient une assomption univariée puisque les différentes capacités de régression des facteurs sont seulement obtenues à partir de dimensions de codes individuelles.

Nous caractérisons maintenant la cohérence des métriques relativement à une réduction de la compacité entre les sous-scénarios 2) et 3). SAP est très stricte sur la compacité*,* et retourne des mesures nulles à ces sous-scénarios puisque les mêmes coefficients de détermination sont mesurés pour chacune des dimensions de codes représentatives de v_i. Ces mesures nulles sont aussi attribuables à MIG pour une raison similaire à SAP, soit la soustraction de deux relations quantifiées à la même valeur. JEMMIG et DCIMIG retournent une même mesure non-nulle pour les sous-scénarios 2) et 3) démontrant un fonctionnement similaire à *MIG* de leur composant mesurant la compacité. Tant qu'à *DCI Lasso compactness*, des mesures

parfaites sont attribuées aux sous-scénarios 2) et 3), ce qui est problématique. Suite à une inspection des poids de régresseurs obtenus suite à l'entraînement du modèle interne, il est observé que la régularisation Lasso cause l'optimisation de seulement une des deux dimensions de codes représentatives d'un facteur, ce qui cause *DCI Lasso Compactness* à retourner un résultat parfait malgré les représentations en réalité non-compactes.

Des résultats très consistants sur l'ensemble des propriétés sont obtenus avec *DCI RF*. Tout d'abord, la métrique retourne les mêmes résultats de compacité pour les sous-scénarios 1) et 2). De plus, la compacité continue à être quantifiée passé 2 dimensions de codes représentatifs. Finalement, on y observe les comportements désirés pour les mesures *DCI RF* E*xplicitness* et *Modularity*.

3.3.5 Description partielle des facteurs

Ce scénario émule un cas où seulement une fraction des facteurs génératifs sont connus, mais l'entièreté est présente dans l'espace de représentation obtenue. Une évaluation partielle des facteurs est typiquement effectuée lorsque des données provenant du monde réel sont récolletées et que seulement l'information d'intérêt à l'application est annotée. Dans le cas où une représentation parfaite de ces données sont obtenues, il est important de s'assurer que les dimensions de codes capturant de l'information impertinente n'affectent pas les mesures démêlage. Une autre possibilité est une représentation contenant des codes faibles ou morts résultant d'une contrainte. Ce scénario évalue donc la robustesse des métriques aux dimensions de codes impertinentes, une propriété décrite à la section 3.3.5.

Pour ce scénario, la relation entre les codes/facteurs est la fonction identité. Les métriques sont ici caractérisées par leur stabilité face à une variation du nombre de facteurs ignorés. Du point de vue des métriques, les codes représentant des facteurs ignorés seront équivalents à du bruit, aussi nommés des *irrelevant-codes* (Eastwood & Williams, 2018). Le comportement désiré est donc d'obtenir les mêmes résultats pour une métrique, peu importe

la fraction de facteurs évalués. La Figure 3.7 représente les résultats obtenus pour ce scénario.

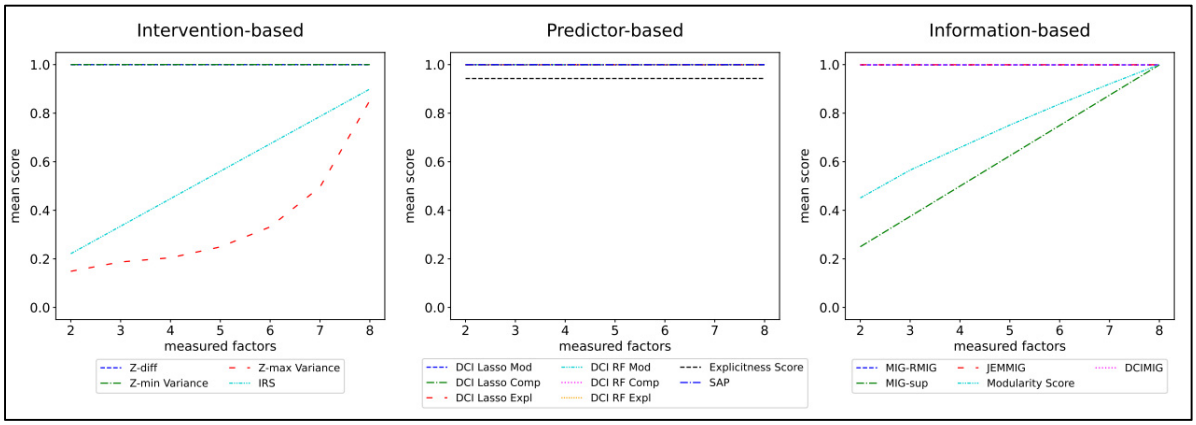

Figure 3.7 Résultats obtenus pour le scénario de description partielle des facteurs (Zaidi, Boilard, Gagnon, & Carbonneau, 2020)

La majorité des métriques se comporte tel que désiré, sauf pour *IRS*, *Z-max Variance*, *MIG-sup* et *Modularity Score*, ce qui limite leur utilité dans des contextes réels. Plusieurs métriques calculent un résultat global en effectuant la moyenne des résultats obtenus pour chacun des codes. Cela est problématique dans ce scénario pour *IRS*, *MIG-sup* et *Modularity score*, puisque ces métriques n'intègrent aucun mécanisme supplémentaire permettant d'ignorer la contribution des *irrelevant-codes*. *Z-max variance* quant à lui fait l'assomption qu'un seul code variera fortement relativement aux autres lorsque tous sauf un facteur sont fixés. Cette assomption est invalide pour ce scénario puisque les *irrelevant-codes* ne sont pas affectés par les interventions sur les facteurs.

3.4 Résumé des interprétations

Un résumé des interprétations effectuées est donné dans le Tableau 3.2. Afin qu'une métrique possède une caractéristique désirée (✓), cela doit être vrai en théorie et se faire sans erreurs en pratique dans nos expériences. Par exemple, *DCI Lasso Compactness* est annoté ×

puisqu'une erreur significative de quantification de la compacité est observée dans les résultats du scénario 3.3.4 « Modulaire, mais non-compacte ». L'annotation « - » indique que la propriété du démêlage associée à la colonne n'est pas mesurée par la métrique, et l'annotation « ×* » indique que l'incompatibilité en question est associée à la discrétisation plutôt que provenant des mécanismes internes de la métrique.

Tableau 3.2 Résumé de l'interprétation des résultats obtenus pour chacun des scénarios

Métrique	Modularité cohérente	Compacité cohérente	Qualité explicite cohérente	Calibrée	Relations non-linéaires	Relations à plusieurs variables indépendantes	Robuste à l'induction de bruit	Robuste aux facteurs non-mesurés	Hyperparamètres stables	Sans discrétisation
Z-diff	×	-	-	✓	×*	✓	✓	✓	×	×
Z-min Variance	×	-	-	✓	×*	✓	✓	✓	×	×
Z-max Variance	×	-	-	×	×*	✓	×	×	×	×
IRS	✓	-	✓	×	×*	n/a	n/a	×	✓	×
DCI - Lasso	✓	×	✓	×	×	✓	✓	✓	✓	✓
DCI – Random Forest	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Explicitness Score	-	-	✓	×	×	✓	n/a	✓	✓	×
SAP	-	✓	-	✓	×	×	n/a	✓	✓	✓
MIG		✓	-	✓	×*	×	×	✓	✓	×
MIG-sup	✓	-	-	✓	×*	×	×	×	✓	×
JEMMIG	✓	✓	✓	✓	×*	×	×	✓	✓	×
Modularity Score	×	-	-	×	×*	×	✓	×	✓	×
DCIMIG	✓	×	✓	✓	×*	×	×	✓	✓	×

Il a été découvert que la majorité des métriques ont des comportements inadéquats en présence d'une relation non-linéaire. Autres que la présence de mécanismes explicitement incompatibles tels que l'utilisation de modèles linéaires, ceux-ci incluent des mécanismes plus subtilement affectés par la non-linéarité, tels qu'une préférence pour des réalisations discrètes à distribution uniforme et/ou la dépendance à des estimations statistiques locales sur

de potentielles faibles populations pour les familles de métriques *intervention-based* et *information-based*.

De plus, il a été observé que plusieurs métriques sont incapables de considérer des relations décrites par plusieurs dimensions de codes indépendantes. Dans le cas de plusieurs d'ensembles de données réels, les facteurs identifiés ne sont pas nécessairement indécomposables et peuvent être confondants de plusieurs sous-facteurs potentiellement découvrables. À noter qu'IRS est exclue de cette caractérisation puisque l'algorithme décrit par les auteurs permet de configurer des sous-ensembles de dimensions de codes pour l'évaluation. Une méthode pour construire des sous-ensembles idéaux est manquante, ce qui empêche de considérer cette possibilité correctement.

Il a aussi été observé que certaines métriques fournissent des mesures sensibles au bruit. Cela les rend susceptibles aux impuretés dans une représentation issue de données à dimensions très élevée et distillée à l'intérieur d'une représentation compacte. À noter que cette caractérisation ne nécessite pas que les métriques de qualité explicite soient robustes à l'induction de bruit.

Lorsque des dimensions de codes supplémentaires sont impertinentes à la description des facteurs d'intérêt, il est important que cela ne biaise pas les mesures obtenues. Il a été observé que plusieurs mesures agglomèrent fautivement ces dimensions supplémentaires sans considération à leur représentativité nulle. Ces métriques seraient donc incapables de prendre en considération les cas pratiques où des dimensions de codes sont régularisées de façon à compacter l'information jusqu'à même désactiver certaines dimensions de codes (Higgins, et al., 2017) (Kim & Mnih, 2018) (Chen, Li, Grosse, & Duvenaud, 2018) et/ou encouragées à séparer les composantes principales découvertes pouvant également inclure les facteurs d'intérêt avec d'autres facteurs non-identifiés (Rolinek, Zietlow, & Martius, 2019).

De plus, la caractérisation effectuée inclut un aperçu des problèmes relatifs aux hyperparamètres de chacune des métriques. Certaines métriques contiennent des hyperparamètres sensibles malgré la faible complexité des représentations générées par les scénarios. Finalement, la granularité de discrétisation est un hyperparamètre relatif au prétraitement des données qui s'est avéré souvent problématique. Toutes métriques nécessitant une discrétisation à bornes également distancées sont automatiquement pénalisées lorsque des relations non-linéaires à valeurs continues sont présentes. Un autre problème associé à la discrétisation est que la granularité idéale peut varier d'un ensemble de données à l'autre, d'une représentation obtenue à l'autre et d'une dimension de code. Aucun mécanisme n'est inclus pour optimiser la méthode de discrétisation. La prudence est recommandée pour les métriques nécessitant la discrétisation des facteurs, et doublement pour les métriques nécessitant en plus la discrétisation des codes.

En général, *DCI Random Forest* est la métrique la mieux adaptée à l'ensemble des cas identifiés dans cette étude Tout d'abord, cette métrique retourne un résultat pour chacune des propriétés quantifiables à l'aide de l'entraînement d'un seul modèle. L'algorithme *Random Forest* est capable de découvrir des relations riches et est la seule qui ne pénalise pas les relations non-linéaires. *DCI RandomForest* est de plus polyvalent dans le sens qu'une mesure globale peut être obtenue pour une composition de facteurs continus et catégoriques. Il suffit de configurer la métrique d'entraîner un classificateur ou un régresseur pour chacun des facteurs. Les mécanismes internes de DCI permettent d'obtenir des mesures robustes aux dimensions de codes ne représentant aucun des facteurs identifiés. Cependant, cette métrique contient des hyperparamètres pouvant grandement influencer les mesures obtenues. Pour contrer cela, les auteurs proposent d'optimiser la profondeur maximale des arbres sur un ensemble de validation. **Pour contrer cela, nous suggérons plutôt que les hyperparamètres d'importance au sur/sous-apprentissage soit optimisés sur un banc de validation croisée.** Dans nos implémentations, nous optimisons les hyperparamètres de profondeur maximale des arbres et le nombre de paramètres maximaux considérés

aléatoirement pour chacun des nœuds. Le désavantage de cette métrique devient donc qu'il s'agit de la métrique dont les mesures prennent le plus de temps à calculer.

CHAPITRE 4

DISCUSSION

Ce chapitre discute de problèmes observés lors de la caractérisation des métriques. Des dépendances observées entre différentes propriétés du démêlage dans nos scénarios sont établies. Des différences entre les environnements expérimentaux d'évaluation les ensembles de données réels est soulignée afin de définir de bonnes pratiques adaptées à ces différences. Des faiblesses de métriques potentiellement surmontables sont soulignées afin d'ouvrir une discussion sur de potentielles améliorations. Finalement, une stratégie afin de rapporter les mesures dans de futurs ouvrages est recommandée.

4.1.1 Relation entre les différentes propriétés du démêlage

Différentes propriétés du démêlage sont présentées à la section 1.1. Celles-ci peuvent être mesurées séparément, mais me sont pas indépendantes. Les dépendances observées dans nos scénarios sont mises en évidence dans cette section.

Toute représentation modulaire et/ou compacte est au minimum légèrement explicite. En effet, il serait impertinent de mesurer le démêlage avec une représentation ne contenant aucune information relativement aux facteurs. Tel que présenté à la section 3.3.1, une représentation peut être mesurée comme fortement modulaire/compacte même pour les représentations où la qualité explicite est basse. Ainsi, l'équation 5.1 implique qu'une qualité explicite non-nulle est une condition nécessaire à l'obtention d'une représentation compacte et/ou modulaire, mais que la qualité mesurée de ces propriétés n'est pas informative de la qualité explicite.

La compacité est une propriété dépendante de la dimensionnalité de la représentation. Supposons le cas où une représentation contient des codes d'une même dimensionnalité d que le nombre de facteurs identifiés M et est complètement explicite de tous les facteurs. Lorsque cette représentation est en plus parfaitement modulaire, cela implique sans

équivalence que la représentation est aussi parfaitement compacte (Eq. 5.2). Cependant, lorsque la dimensionnalité d augmente, cela rend possible la dégradation de la compacité. Dans ce cas, une représentation parfaitement modulaire implique seulement que la compacité est potentiellement parfaite. Si la compacité reste parfaite, les dimensions de codes supplémentaires sont implicitement impertinentes relativement aux facteurs mesurés. Le potentiel de dégradation de la compacité est plus élevé lorsque la dimensionnalité des codes *d* est de plus en plus grand relativement au nombre de facteurs M. Tout de même, il est idéal que $d > M$ afin de permettre l'obtention d'une représentation polyvalente permettant des facteurs non-identifiés à être représenté, en plus de permettre les facteurs d'intérêt à être décrits par une composition de dimensions de codes. Le potentiel de dégradation de la compacité est défini comme une propriété dans l'équation 5.3.

Le Tableau 4.1 met en proposition par implication les comportements décrits ci-haut. Les indices k indiquent une représentation quelconque. Les variables Mod_k, $Comp_k$ et $Expl_k$, représentent respectivement la qualité des propriétés de *Modularity*, *Compactness* et *Explicitness* d'une quelconque k'ème représentation, et d_k représente la dimensionnalité des codes de cette représentation.

Tableau 4.1 Mise en proposition de la dépendance des propriétés du démêlage

$A_k : (Mod_k = 1) \wedge (Expl_k = 1)$	
$(Mod_k > 0) \vee (Comp_k > 0) \rightarrow (Expl_k > 0)$	(4.1)
$A_k \wedge (d_k = M) \rightarrow (Comp_k = 1)$	(4.2)
$A_1 \wedge A_2 \wedge (d_1 > d_2) \wedge (d_2 \geq M) \rightarrow Comp_1 \leq Comp_2$	(4.3)

4.1.2 Données jouets vs Données réelles

Il est d'abord important de différencier les ensembles de données jouets des ensembles de données réels. Dans les ensembles « réels », les facteurs sont identifiés plutôt que d'être des variables indépendantes configurant la génération d'observations. Ainsi, l'indépendance et la non-corrélation des facteurs identifiés ne sont pas assurées dans un ensemble de données réelles. Par exemple, on pourrait s'intéresser à obtenir une représentation d'un ensemble de données de fruits où les caractéristiques de catégories de fruits et couleur de fruits sont démêlées. Dans cet exemple, toutes mesures de compacité et modularité seront biaisées. Le problème avec cette approche est qu'une catégorie de fruit ne cooccurre réalistement jamais avec certaines couleurs de fruits et davantage avec d'autres. Ainsi, un biais pour les mesures de modularité est inévitable puisque les dimensions de codes représentatives de la catégorie du fruit seront intrinsèquement représentatives de la couleur, et un biais pour les mesures de compacité est inévitable puisque la prédiction de la couleur pourra effectuer ses prédictions à l'aide des deux dimensions de codes qui seront représentatifs des deux facteurs.

De plus, il est important de considérer que les annotations d'ensembles réels ne sont pas exhaustives. Par exemple, le fruit peut reposer sur une surface quelconque ou se trouver dans un arbre. La couleur de ces éléments arrière-plan peut varier, et certaines ombres pourraient être présentes dans les images. Lorsqu'un modèle est entraîné avec une fonction objective sans biais inductifs relativement aux facteurs d'intérêt, le processus d'entraînement considérera les autres facteurs avec la même importance. De plus, les facteurs d'intérêt peuvent aussi être corrélés avec ces facteurs non-identifiés. Il est donc impératif que les métriques utilisées soient capables de faire abstraction de facteurs non-identifiés afin de mesurer la représentation relativement aux facteurs d'intérêt.

Les mécanismes génératifs des ensembles jouet permettent facilement l'homogénéité d'une classe et l'uniformité de leurs distributions, ce qui est obtenu implicitement pendant le processus génératif. Dans SmallNORB (LeCun, Huang, & Bottou, 2004), l'ensemble des

observations sont générées à partir d'un facteur catégorique et 3 facteurs discrets qui pourraient être décrits dans un domaine continu, soit 6 valeurs d'intensité d'éclairage, 9 valeurs d'élévation et 18 valeurs d'azimuts. Ces valeurs continues sont prédiscrétisées. Par exemple, les 6 valeurs possibles d'intensité d'éclairage correspondent à {0, 0.2, 0.4, …, 1}. L'homogénéité d'une classe est donc assurée. De plus, afin d'assurer une distribution uniforme des facteurs, une observation est générée pour chacune des combinaisons possibles de classes. Ainsi, dans le monde réel, l'obtention de cette uniformité et homogénéité n'est pas assurée par la discrétisation d'annotations de facteurs continus. Ainsi, toutes métriques nécessitant une discrétisation s'expose à de potentiels cas d'erreurs.

Finalement, même avec une granularité de discrétisation générique, tel que dans la librairie *disentanglement-lib* publiée avec (Locatello, et al., 2019), la non-uniformité distributionnelle peut s'avérer assez sévère pour que certaines combinaisons de réalisations discrètes des facteurs et/ou codes n'existent pas. Cela peut causer des biais de mesures, où même des cas d'erreurs pour les métriques nécessitant une forte population de combinaisons différentes.

4.1.3 Considérations pratiques

Dans les cas pratiques, prioriser la modularité plutôt que la compacité permet la sélection de représentations plus polyvalentes, tel que discuté dans (Ridgeway & Mozer, 2018). Il faut considérer qu'une mesure de compacité sera seulement fiable lorsque des facteurs complètement décomposés sont identifiés. Par exemple, la luminosité d'une image pourrait contenir plusieurs attributs, tels que la couleur, la forme, l'intensité, l'élévation et l'angle de multiples sources lumineuses. Si de telles décompositions ne sont pas identifiées et qu'une méthode d'apprentissage obtient une représentation ayant découvert de tels sous-facteurs, prioriser la compacité peut causer le praticien à ignorer d'excellentes représentations. Prioriser la modularité comporte moins de désavantages dans ce cas. Une représentation reste tout de même modulaire même si de tels sous-facteurs sont découverts. Ainsi, le praticien prend en considération qu'une bonne représentation peut contenir

n'importe quelle décomposition potentiellement découvrable d'un facteur afin de le représenter. Ainsi, prioriser la compacité est seulement désirable lorsque l'on peut identifier des facteurs indécomposables et dont la capacité à décrire l'ensemble des données est absolue.

Les métriques incapables de faire abstraction des facteurs non-identifiés devraient être évitées dans un contexte réel. Il est difficile, voire pratiquement impossible, d'identifier l'ensemble des facteurs possibles dans un ensemble de données réel. Ainsi, une bonne métrique pour un contexte pratique doit être capable de faire abstraction des facteurs non-identifiés. Rappelons que toute information relative à des facteurs non-identifiés peut être vue comme du bruit. Les mesures de modularité et compacité perdant leur cohérence suite à une induction de bruit (3.3.1) ainsi que celles affectées par l'action d'ignorer des facteurs (3.3.5) devraient être évitées.

La discrétisation des codes et/ou des facteurs n'est pas une opération triviale. Comme observé dans le scénario non-linéaire de la section 3.3.3, les métriques nécessitant une discrétisation des facteurs sont toutes négativement affectées lorsque l'intensité de la non-linéarité s'accentue. Dans ce cas-ci, cela est parce que les métriques se comportent généralement mieux lorsque les réalisations discrètes de valeurs continues sont d'une distribution uniforme. Puisque la discrétisation est généralement effectuée à l'aide de bornes également distancées, une distribution uniforme peut seulement être assurée si la relation est linéaire. D'autres cas d'erreurs peuvent aussi se manifester en fonction de la granularité de discrétisation choisie. Dans les cas des métriques intervention-based, une granularité trop grossière cause une trop grande hétérogénéité dans un groupe assumé être la même valeur pour la population entière et une discrétisation fine réduit la variété des sous-ensembles possibles à échantillonner. Dans le cas des métriques *information-based*, la granularité a beaucoup d'effets indésirables. Une discrétisation trop grossière cause une surestimation de l'information mutuelle en augmentant la cooccurrence des classes. Une granularité fine rend les estimations de l'information mutuelle plus susceptibles à l'induction de bruit, et une

granularité trop fine surestime l'information mutuelle en créant beaucoup de co-occurrences exclusives.

4.1.4 Recommandations futures pour la mesure du démêlage

Les mesures devraient être observées individuellement à chacun des facteurs/codes plutôt que par le biais d'un aperçu global tel qu'une moyenne. Les mesures moyennes donnent un aperçu global de la qualité des propriétés mesurées. Cependant, il n'est pas possible de discerner à partir de la moyenne si une seule des mesures individuelles où l'ensemble des mesures sont responsables d'une mauvaise mesure globale. Cela aide aussi au déverminage de mauvaises mesures. Une mauvaise mesure d'un facteur spécifique sur une grande quantité de représentations permet l'identification d'un facteur problématique, alors que de mauvaises mesures globales pourraient signifier la mauvaise performance du modèle utilisé pour obtenir une représentation.

Les mesures devraient être calculées de nombreuses fois sur une même représentation, et être rapportées selon leurs moyennes et leur déviation standard. Plusieurs métriques implémentent des mécanismes aléatoires. Par exemple, l'échantillonnage de sous-ensembles dans les métriques *intervention-based*, le passage d'un générateur de nombre aléatoire pour l'entraînement d'un modèle *predictor-based*, ou la fragmentation d'un ensemble de données en ensembles d'entraînement, de validation et de test. De plus, lorsqu'un ensemble de données est d'une grande population, il est commun d'évaluer le démêlage avec seulement un sous-ensemble de celui-ci pour des raisons d'efficacité. Ces processus stochastiques peuvent faire varier les mesures obtenues, et ce malgré la stabilité de la métrique. Nous suggérons donc de rapporter les résultats de façon à assurer leur signifiance face à leur variance.

Afin de motiver une potentielle future amélioration aux métriques existantes, rappelons qu'**une représentation modulaire et compacte est seulement possible lorsque les**

dimensions de codes sont parfaitement alignées avec les facteurs. Supposons le cas où une représentation des facteurs parfaitement compacte et modulaire des facteurs est apprise. Une rotation réversible entre les différents axes de la représentation est effectuée et la représentation demeure tout aussi explicite et conserve ses composantes principales. Malgré cela, toutes mesures de modularité ou compacité retourneront des mesures inférieures. Il serait idéal qu'une métrique soit robuste au « choix du point de vue » sur la représentation. Jusqu'à maintenant, aucune métrique n'incorpore de mécanismes capables d'offrir une robustesse à de tels cas.

Sinon, quant aux problématiques reliées à la discrétisation, toutes les métriques en nécessitant l'effectuent avec un nombre fixe de bornes uniformément espacées, et aucune solution cherchant à optimiser les bornes de discrétisation n'a été proposée. Tant qu'aucune amélioration à cet effet n'est effectuée, l'interprétabilité des métriques des familles *intervention-based* et *information-based* est pénalisée.

CONCLUSION

Pour toutes métriques de nature quelconque, le praticien doit être capable d'identifier exactement ce qu'elles mesurent. Plusieurs métriques sont proposées dans la littérature afin de mesurer le démêlage. La taxonomie définie selon 3 familles, soit *intervention-based*, *predictor-based* et *information-based*, permet de mieux regrouper les métriques afin que le praticien puisse mieux organiser ses connaissances. De plus, elle permet d'exposer plusieurs similarités entre les métriques. Ces similarités ne sont cependant pas absolues, comme démontré par le manque de corrélation entre les métriques d'une même famille et/ou mesurant les mêmes propriétés. Cela a davantage motivé l'étude des métriques afin de comprendre les désaccords provenant de différents mécanismes internes.

Afin de démystifier les désaccords et interprétations de métriques, un banc de test éliminant des incertitudes ciblées quant à la nature de représentations a été proposé. Celui-ci génère des représentations spécialisées afin de mesurer des propriétés précises. Ainsi, plusieurs sources d'erreurs et plusieurs erreurs provenant de mécanismes de métriques ont été découvertes et rendues explicites, telles que la discrétisation, la non-linéarité d'une relation code-facteur et la présence de dimensions de codes superflues.

Il faut de plus considérer que la mesure du démêlage est habituellement effectuée dans des environnements expérimentaux où toutes les conditions sont idéales. Jusqu'à maintenant, la notion de démêlage ainsi que les algorithmes implémentés et ouvertement accessibles étaient plutôt réservés aux environnements expérimentaux. Ainsi, en plus de proposer des implémentations plus polyvalentes, cet ouvrage a discuté de plusieurs considérations pratiques difficiles à éviter dans des ensembles réels, soit en exposant des défauts qui ne sont habituellement pas rencontrés dans des environnements expérimentaux.

Considérant les cas d'applications génératifs motivant l'étude de la mesure du démêlage, les résultats obtenus nous amènent à suggérer que le praticien obtient une multitude de représentations et filtre ceux-ci à l'aide de la métrique DCI RandomForest, puisqu'elle

générera des mesures polyvalentes avec le minimum de risques d'erreur. Ensuite, si le praticien s'intéresse seulement à une fraction des propriétés jugées idéales de cet ouvrage, celui-ci est invité à consulter le Tableau 3.2 et sélectionner des métriques qui sont au minimum cohérentes avec les propriétés qu'il jugera importantes. À noter qu'il est donc tout de même recommandé de procéder à un certain point à une évaluation humaine et subjective des exemples générés. Cela permettra de valider d'autres éléments plus difficiles à mesurer quantitativement, tel que la cohérence de tous autres éléments présents autres que les facteurs identifiés.

À titre d'amélioration, aucune métrique de modularité/compacité n'est adaptée au cas de la rotation des composantes principales d'une représentation parfaite, tel que décrit à la section 4.1.4. La capacité de considérer cette rotation permettrait d'identifier le démêlage des facteurs relativement aux composantes principales individuelles d'une représentation plutôt que relativement aux dimensions des codes. Malheureusement, tant et aussi longtemps que ce cas n'est pas adressé, le praticien faisant usage des métriques de démêlage pour justifier la sélection d'une représentation est à risque de passer à côté de plusieurs représentations idéales. Finalement, il serait possible d'obtenir de meilleures performances pour les métriques des familles *intervention-based* et *information-based* si des ajustements bénéfiques sont apportés à la discrétisation.

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

Abdi, A. H., Abolmaesumi, P., & Fels, S. (2019). A preliminary study of disentanglement with insights on the inadequacy of metrics. *ArXiv:1911.11791*.

Balasubramanian, V., Kobyzev, I., Bahuleyan, H., Shapiro, I., & Vechtomova, O. (2020). Polarized-vae: Proximity based disentangled representation learning for text generation. *arXiv preprint*, arXiv:2004.10809.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence, 35*, 1798-1828.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). *Classification and regression trees.* Routledge.

Chen, R. T., Li, X., Grosse, R., & Duvenaud, D. (2018). Isolating sources of disentanglement in variational autoencoders. *Advances in Neural Information Processing Systems.*

Do, K., & Tran, T. (2020). Theory and evaluation metrics for learning disentangled representations. *ICLR*.

Duan, S., Matthey, L., Saraiva, A., Watters, N., Burgess, C., Lerchner, A., et al. (2020). Unsupervised model selection for variational disentangled representation learning. *International Conference on Learning Representations*.

Eastwood, C., & Williams, C. K. (2018). A framework for the quantitative evaluation of disentangled representations. *ICLR.*

Eyben, F., Scherer, K., Schuller, B., Sundberg, J., André, E., Busso, C., et al. (2015). The geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing. *IEEE transactions on affective computing, 7*(2), 190-202.

Guo, W., Wang, J., & Wang, S. (2019). Deep multimodal representation learning: A survey. *IEEE Access*, 63373-63394.

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., et al. (2018). Towards a definition of disentangled representations. *arXiv:1812.02230*.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., et al. (2017). β-VAE: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations.*

Kim, H., & Mnih, A. (2018). Disentangling by factorising. *International Conference on Machine Learning*, (pp. 4153-4171).

Kim, M., Wang, Y., Sahu, P., & Pavlovic, V. (2019). Relevance Factor VAE: Learning and identifying disentangled factors. *arXiv:1902.01568*.

Kingma, D. P., & Welling, M. (2013). Auto-Encoding variational bayes. *arXiv:1312.6114*.

Kumar, A., Sattigeri, P., & Balakrishnan, A. (2018). Variational inference of disentangled latent concepts from unlabeled observations. *International Conference on Learning Representations.*

LeCun, Y., Huang, F. J., & Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2*, pp. II-104.

Li, Y., Singh, K. K., Ojha, U., & Lee, Y. J. (2020). Mixnmatch: Multifactor disentanglement and encoding for conditional image generation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 8039-8048).

Li, Z., Murkute, J. V., Gyawali, P. K., & Wang, L. (2020). Progressive learning and disentanglement of hierarchical representations.

Liu, X., Thermos, S., Valvano, G., Chartsias, A., O'Neil, A., & Tsaftaris, S. A. (2020). Metrics for exposing the biases of content-style disentanglement. *arXiv:2008.12378*.

Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., et al. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. *International Conference on Machine Learning*.

Mahony, N. O., Campbell, S., Carvalho, A., Harapanahalli, S., Velasco-Hernandez, G., Krpalkova, L., et al. (2019). Deep learning vs. traditional computer vision. *arXiv:1910.13796*.

Paier, W., Hilsmann, A., & Eisert, P. (2020). Interactive facial animation with deep neural networks. *IET Computer Vision, vol. 14*(6), 359-369.

Reed, S. E., Zhang, Y., Zhang, Y., & Lee, H. (2015). Deep visual analogy-making. *Advances in Neural Information.*

Ridgeway, K. (2016). A Survey of inductive biases for factorial representation-learning. *arXiv:1612.05299*.

Ridgeway, K., & Mozer, M. (2018). Learning deep disentangled embeddings with the F-statistic loss. *Advances in Neural Information Processing Systems.*

Rolinek, M., Zietlow, D., & Martius, G. (2019). Variational autoencoders pursue PCA directions (by accident). *IEEE Conference on Computer Vision and Pattern Recognition*.

Sepliarskaia, A., Kiseleva, J., & de Rijke, M. (2020). Evaluating disentangled representations. *arxiv:1910.05587*.

Suter, R., Miladinović, D., Schölkopf, B., & Bauer, S. (2019). Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. *ICML.*

Wang, Y., Stanton, D., Zhang, Y., Skerry-Ryan, R. J., Battenberg, E., Shor, J., et al. (2018). Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. *Internation Conference on Machine Learning*, (pp. 5180-5189).

Zaidi, J., Boilard, J., Gagnon, G., & Carbonneau, M.-A. (2020). Measuring disentanglement: A review of metrics. *arXiv:2012.09276*.