Dissimilarités entre jeux de données

William Raynaut, Chantal Soule-Dupuy, Nathalie Valles-Parlangeau

IRIT, Université de Toulouse Toulouse, France

prenom.nom@irit.fr

RÉSUMÉ. *La caractérisation de jeu de données reste un obstacle majeur pour l'analyse de données intelligente. Nombre d'approches à ce problème agrègent les informations décrivant les attributs individuels des jeux de données, ce qui représente une perte d'information. Nous proposons une approche par dissimilarité permettant d'éviter cette agrégation, et étudions son intérêt dans la caractérisation des performances d'algorithmes de classification et dans la résolution de problèmes de méta-apprentissage.*

ABSTRACT. *Characterizing datasets has long been an important issue for algorithm selection and meta-level learning. Most approaches share a potential weakness in the aggregation of informations about individual features of the datasets. We propose a dissimilarity based approach avoiding this particular issue, and show the benefits it can yield in characterizing the appropriateness of classification algorithms, and in the context of meta-level classification.*

MOTS-CLÉS : *caractérisation de jeux de données, dissimilarité, méta-attributs, sélection d'algorithmes, méta-apprentissage.*

KEYWORDS: *dataset characterization, dissimilarity, meta-features, meta-learning, algorithm selection.*

DOI:10.3166/ISI.22.3.35-63

1. Motivation

1.1. Introduction

L'émergence du phénomène de données massives crée un besoin grandissant en analyse de données, et bien souvent, cette analyse est conduite par des experts de différents domaines ayant peu d'expérience en science des données. Afin de leur permettre de tout de même exploiter efficacement leurs données, divers travaux ont proposé des méthodes d'assistance intelligente à l'analyse de données (Zakova *et al.*, 2011 ; Serban, 2013). La caractérisation de jeu de données, problème apparu avec les premières ébauches de méta-apprentissage (Giraud-Carrier *et al.*, 2004), constitue encore l'un des verrous majeurs de l'assistance intelligente à l'analyse de données. L'enjeu de la caractérisation de jeu de données repose sur une des hypothèses fondamentales de la méta-analyse : *Si un algorithme d'analyse Φ a démontré de bonnes performances sur un jeu de données A et que le jeu de données B est* ***similaire*** *à A, alors l'algorithme Φ aura probablement de bonnes performances sur B*. Être capable de comparer efficacement la topologie de jeux de données permet alors d'exploiter la connaissance que l'on peut avoir sur la performance d'algorithmes d'analyse sur différents jeux de données pour choisir des méthodes adaptées lors de futures analyses.

Dans le cadre particulier du méta-apprentissage, le problème de caractérisation de jeu de données consiste en la définition d'un ensemble de propriétés de jeu de données (ou méta-attributs) permettant leur caractérisation précise, qui doit de plus être utilisable par des algorithmes de méta-apprentissage. Afin de se conformer aux prérequis de la plupart des algorithmes de méta-apprentissage, ces propriétés sont généralement agrégées en vecteurs d'attributs de taille fixe, ce qui peut représenter une importante perte d'information (Kalousis, Hilario, 2001b). Nous étudions la possibilité que les limitations des techniques courantes de caractérisation de jeu de données soient l'un des obstacles majeurs à la bonne performance de la sélection d'algorithmes. Nous nous concentrons en particulier sur la définition d'une représentation des jeux de données permettant d'utiliser toute l'information disponible pour leur caractérisation.

1.2. Approches précédentes

On peut distinguer deux catégories d'approches au problème de caractérisation de jeux de données :

– Le premier consiste en l'emploi de mesures statistiques et information-théorétiques pour décrire le jeu de données. Cette approche, notamment mise en avant par le projet STATLOG (Michie *et al.*, 1994), et employée dans une majorité d'études postérieures (Vilalta, Drissi, 2002 ; Kalousis *et al.*, 2004 ; Leite *et al.*, 2012 ; Sun, Pfahringer, 2013 ; Leyva *et al.*, 2015), présente nombre de mesures très expressives, mais sa performance repose intégralement sur l'adéquation entre le biais de l'apprentissage effectué au méta-niveau et l'ensemble de mesures choisies. On note parfois l'emploi de techniques de sélection d'attributs à ce méta-niveau (Kalousis, Hilario, 2001a), mais les résultats expérimentaux ne permettent pas de conclure à la supériorité d'une

quelconque mesure indépendamment du méta-apprentissage employé (Todorovski *et al.*, 2000).

– Le second axe d'approche considère quant à lui non pas des propriétés intrinsèques du jeu de données étudié, mais plutôt la performance d'algorithmes d'apprentissage simples exécutés dessus. Introduit comme «*landmarking*» par (Pfahringer *et al.*, 2000), cette approche emploie initialement le taux d'erreur d'un ensemble d'algorithmes basiques comme métadonnées. Comme précédemment, les résultats suggèrent une forte dépendance de l'efficacité de cette approche avec le choix des algorithmes de base et du méta-niveau, ne révélant aucune combinaison uniformément supérieure. Des développements postérieurs ont introduit des mesures plus complexes, tel (Peng *et al.*, 2002) proposant comme méta-attributs des propriétés structurelles d'un arbre de décision construit sur la donnée. Les expériences conduites par (Fürnkranz, Petrak, 2002) sur ces différentes approches tendent à conclure que toutes peuvent réaliser de bonnes performances dans diverses parties de l'ensemble des jeux de données, sans qu'aucune ne domine globalement.

Le problème de caractérisation de jeux de données a donc déjà reçu une certaine attention dans le domaine du méta-apprentissage, mais l'agrégation des méta-attributs en vecteur de taille fixe y reste une constante. Cette agrégation représente cependant une importante perte d'information, que certaines approches ont déjà tenté de limiter, notamment par l'utilisation d'histogrammes (Kalousis, 2002). On peut illustrer ce problème sur l'exemple suivant.

1.3. Exemple

Considérons deux jeux de données, **A** et **B** illustrés en figure 1. **A** décrit 12 attributs de 100 individus, et **B** 10 attributs de 200 individus. On souhaite comparer les résultats de 5 mesures statistiques et informationnelles relevées sur les attributs individuels de ces jeux de données (comme illustré sur le second attribut de **A**).

L'information complète que l'on souhaite comparer est donc un vecteur de 60 valeurs pour **A** et de 50 pour **B**. Une approche classique (Kalousis, 2002 ; Wistuba *et al.*, 2015) serait de faire une moyenne de chaque méta-attribut selon les différents attributs des jeux de données, perdant ainsi l'information caractérisant individuellement chaque attribut (figure 2a)).

Notre approche est de comparer les attributs de **A** et **B** par paires les plus similaires, identifiant les attributs en surnombre à d'hypothétiques attributs vides. L'hypothèse émise ici est qu'un attribut absent équivaut à un attribut dont aucune valeur n'est connue. Pour en revenir à l'exemple, la comparaison des 5 mesures s'effectuera donc entre l'attribut de **A** et l'attribut de **B** les plus similaires *selon ces mêmes mesures*, puis sur les seconds plus similaires et ainsi de suite, pour finir par comparer les mesures relevées sur les deux attributs surnuméraires de **A** avec leur valeur sur un hypothétique attribut vide de **B**. Cette comparaison par paires permet de s'affranchir de

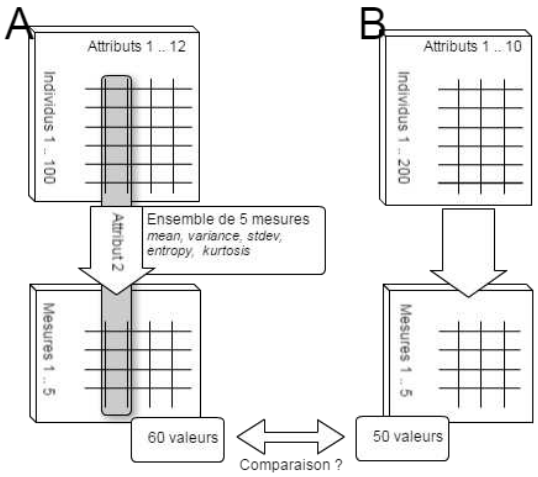
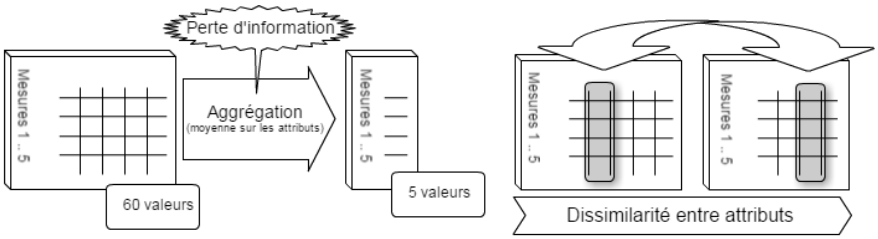


Figure 1. Propriétés d'attributs individuels

l'ordre de présentation des attributs, qui ne recèle aucune information, se concentrant sur la topologie réelle du jeu de données (figure 2b).



(a) Moyenne sur les attributs (b) Comparaison directe d'attributs

Figure 2. Comparaison d'attributs évitant la perte d'informations

Cette comparaison peut s'effectuer à l'aide d'une dissimilarité prenant en compte les propriétés des attributs individuels des jeux de données.

2. Fonction de dissimilarité

2.1. Propriétés désirables

Avant de proposer une fonction candidate, il convient d'étudier les propriétés qu'elle devrait présenter. Soit Ω l'ensemble des jeux de données, et $x, x' \in \Omega$ des instances de jeu de données. Traditionnellement, les propriétés usuelles des distances

ne sont pas jugées nécessaires (Wang *et al.*, 2009), ne conservant que la positivité ($d(x, x') \geq 0$). On préfère cependant conserver uniquement des propriétés présentant une interprétation naturelle dans le contexte de la caractérisation de jeu de données.

$\surd$ Positivité ($d(x, x') \geq 0$) : une dissimilarité doit quantifier la différence absolue entre éléments, donc naturellement positive.

$\surd$ Indiscernabilité des identiques ($x = x' \rightarrow d(x, x') = 0$) : des jeux de données rigoureusement identiques doivent être considérés aussi similaires que possible.

$\times$ Identité des indiscernables ($d(x, x') = 0 \rightarrow x = x'$) : des jeux de données physiquement différents doivent pouvoir être considérés parfaitement similaires (considérer par exemple l'ordre de présentation des attributs).

$\surd$ Symétrie ($d(x, x') = d(x', x)$) : l'ordre de présentation des jeux de données est indifférent, il parait donc naturel de l'ignorer.

$\times$ Inégalité triangulaire ($d(x, x'') \leq d(x, x') + d(x', x'')$) : perd tout sens en l'absence d'identité des indiscernables. On peut avoir $d(x, x') = d(x', x'') = 0$ et néanmoins $x \neq x''$ et $d(x, x'') \geq 0$...

Dï£¡FINITION 1. — *Soit A un ensemble et d une fonction de $A^2 \rightarrow \mathbb{R}$. d est une* ***fonction de dissimilarité*** *sur A si et seulement si, $\forall x, x' \in A^2$:*

- $d(x, x') \geq 0$ *(Positivité)*
- $x = x' \rightarrow d(x, x') = 0$ *(Indiscernabilité des identiques)*
- $d(x, x') = d(x', x)$ *(Symétrie)*

Afin de construire une dissimilarité entre jeux de données, il faudra pouvoir comparer les valeurs de différentes propriétés de ces jeux de données. Ces propriétés étant possiblement très différentes les unes des autres de par leur sémantique et représentation, une normalisation sera nécessaire pour garantir qu'aucune composante ne domine les autres ou ne soit ignorée. Ces propriétés sont formalisées dans la définition ci-dessous, où un ensemble de valeurs sera dit *atomique* s'il ne peut être divisé en sous-ensembles dont l'observation indépendante produirait autant d'information que l'observation de l'ensemble complet.

Dï£¡FINITION 2. — *Soit A un ensemble fini et d une fonction de dissimilarité sur A. d est une* ***fonction de dissimilarité normalisée*** *sur A si et seulement si au moins l'une des propriétés suivantes est vérifiée :*

1. A est atomique et d est bornée sur A

2. Il existe une suite d'ensembles $E_1 ... E_n$ tels que $A = \prod_{i=1}^{n} E_i$, et une suite de fonctions de dissimilarité $d_1 ... d_n$ respectivement normalisées sur $E_1 ... E_n$, telles que :

$$\forall \delta \in \mathbb{R},\ \exists \Delta \in \mathbb{R} \textit{ tel que } \forall i \in [1..n] \textit{ et } \forall a, b, c \in A,$$
$$\textit{Si } d_i(a_i, b_i) = d_i(a_i, c_i) + \delta * \max_{(x,y) \in A^2} (d_i(x_i, y_i))$$
$$\textit{et } \forall j \in [1..n],\ j \neq i,\ d_j(a_j, b_j) = d_j(a_j, c_j)$$
$$\textit{Alors } d(a, b) = d(a, c) + \Delta \textit{ et } \Delta = 0 \leftrightarrow \delta = 0$$

En d'autres termes, une variation d'amplitude δ relativement à sa borne supérieure, de toute composante d_i entre deux éléments de A induit une même variation Δ de d.

2.2. Fonction candidate

Nous proposons ici une fonction de dissimilarité particulière présentant les propriétés énoncées précédemment. Pour construire ces fonctions, nous considérons un ensemble fini de jeux de données ω, et deux ensembles de mesures. Le premier, G, consiste en des propriétés générales de jeu de données, telles que présentées en section 2. Le second, F, consiste en des propriétés capables de caractériser les attributs individuels de jeux de données.

DÏ£¡FINITION 3. — *Soit $E_1...E_n$ une suite d'ensembles finis et A leur produit cartésien $\prod_{i=1}^{n} E_i$. Soit $d_1...d_n$ une suite de fonctions de dissimilarité respectivement sur $E_1...E_n$. On définit la* ***dissimilarité normalisée par la borne supérieure*** *(ou upper bound relative : ubr) sur A selon $d_1...d_n$, $d_A^{ubr} : A^2 \mapsto \mathbb{R}^+$ telle que :*

$$\forall a, b \in A,\ d_A^{ubr}(a,b) = \sum_{i=1}^{n} \frac{d_i(a_i, b_i)}{\max\limits_{(x,y)\in A^2}(d_i(x_i, y_i))} \tag{1}$$

PROPOSITION 4. — *Soit $E_1...E_n$ une suite d'ensembles finis et A leur produit cartésien $\prod_{i=1}^{n} E_i$. Soit $d_1...d_n$ une suite de fonctions de dissimilarité respectivement normalisées sur $E_1...E_n$. Alors, la* ***dissimilarité normalisée par la borne supérieure*** *sur A selon $d_1...d_n$ est une* ***fonction de dissimilarité normalisée*** *sur A.* [1]

Supposant que l'on puisse construire des fonctions de dissimilarité normalisées sur $G(\omega)$ et $F(\omega)$, on pourrait donc proposer d_ω^{ubr} comme fonction de dissimilarité normalisée sur ω. Afin d'alléger les notations, dans les paragraphes suivants, pour toute fonction H définie sur ω, on notera abusivement $d_H(H(x), H(y)) = d_H(x, y)$.

2.2.1. Méta-attributs des jeux de données

Soit un ensemble G de méta-attributs de jeux de données. Les valeurs $g(\omega)$ de l'un de ces méta-attributs g sur nos jeux de données constitueront le cas typique d'ensembles *atomiques* à partir desquels lesquels calculer la dissimilarité. On doit donc définir pour chaque méta-attribut g une dissimilarité bornée $d_g : g(\omega)^2 \mapsto \mathbb{R}^+$ (par exemple la différence absolue), qui selon la définition 2.1 sera donc normalisée. Ceci permet d'introduire la dissimilarité normalisée par la borne supérieure (voir équation (1)) sur $G(\omega)$ selon $\{d_g | g \in G\}$:

$$\forall x, y \in \omega,\ d_{G(\omega)}^{ubr}(x,y) = \sum_{g\in G} \frac{d_g(x,y)}{\max\limits_{(x',y')\in\omega^2}(d_g(x',y'))} \tag{2}$$

1. Voir les *Ressources* en section 5 pour la preuve.

En pratique, cela coïncidera généralement avec une distance de Manhattan normalisée. Cela pose en revanche les fondations nécessaires au prochain type de mesures : les méta-attributs caractérisant les attributs individuels des jeux de données.

2.2.2. Méta-attributs des attributs

Soit un ensemble F de *méta-attributs des attributs* permettant de caractériser les attributs individuels de jeux de données. Certains pourront caractériser tout type d'attribut (le *nombre de valeurs manquantes*, par exemple), tandis que d'autres seront restreints à des types particuliers. Dans la définition de ces méta-attributs, nous considérons les deux types d'attributs les plus représentés : attributs nominaux (prenant un nombre fini de valeurs discrètes) et numériques (prenant valeur dans un espace non fini, souvent $\mathbb{R}$). Les vecteurs de méta-attributs caractérisant les attributs individuels présenteront donc nécessairement des valeurs manquantes (notées $\emptyset$) pour les mesures inadaptées à leur type, ce qui est un obstacle majeur à leur comparaison. En effet, la signification d'une différence de valeur entre deux jeux de données est intrinsèquement dépendante du méta-attribut considéré, et varie grandement d'un méta-attribut à l'autre. Afin de pouvoir comparer la valeur $f(x_i)$ d'un méta-attribut $f \in F$ sur x_i le i^{th} attribut du jeu de données $x \in \omega$, et x'_j le j^{th} attribut du jeu de données $x' \in \omega$, on introduit les fonctions $\delta_f : f(\omega)^2 \mapsto \mathbb{R}^+$ et $\delta_f^\emptyset : f(\omega) \mapsto \mathbb{R}^+$ telles que :

1) δ_f est une dissimilarité bornée sur l'ensemble atomique $f(\omega)$ (donc normalisée)

2) $\delta_f(x_i, \emptyset) = \delta_f(\emptyset, x_i) = \delta_f^\emptyset(x_i)$

3) $\delta_f^\emptyset$ est la *dissimilarité à l'absence de valeur* du méta-attribut f. Elle doit être définie en considérant le *sens* d'une valeur manquante de f, et sera détaillée plus avant par la suite.

On peut alors définir $\delta_F : F(\omega)^2 \mapsto \mathbb{R}^+$:

$$\delta_F(x_i, x'_j) = \sum_{f \in F} \frac{\delta_f(x_i, x'_j)}{\max\limits_{\substack{(y,y') \in \omega^2 \\ (p,q) \in \mathbb{N}^2}} \delta_f(y_p, y'_q)))} \tag{3}$$

δ_F permet de comparer des attributs de différents jeux de données selon les *Méta-attributs des attributs* de F. Cependant, comme illustré précédemment en figure 2b, l'objectif est de comparer ces attributs par paires. Ceci requiert un mapping entre les attributs de deux jeux de données :

DÏ£¡FINITION 5. — *On définit une fonction de mapping σ comme une application associant à une paire de datasets $(x, x') \in \omega^2$, possédant respectivement n et n' attributs, une paire d'applications (σ_1, σ_2) injectives respectivement de $[\![1, n]\!]$ dans $[\![1, n']\!] \cup \{\emptyset\}$ et de $[\![1, n']\!]$ dans $[\![1, n]\!] \cup \{\emptyset\}$, telles que $\forall a \in [\![1, n]\!]$, et $b \in [\![1, n']\!]$, $(\sigma_1(a) = b) \Leftrightarrow (\sigma_2(b) = a)$. (Voir figure 3a)*

Étant donné une fonction de mapping σ, on peut définir $d^\sigma_{F(\omega)} : F(\omega)^2 \mapsto \mathbb{R}^+$ telle que :

$$d^{\sigma}_{F(\omega)}(x,x') = \frac{1}{max(n,n')}\Big(\sum_{\sigma_1(i)=j}^{i,j}\delta_F(x_i,x'_j) + \sum_{\sigma_1(i)=\emptyset}^{i}\delta_F(x_i,\emptyset) + \sum_{\sigma_2(j)=\emptyset}^{j}\delta_F(\emptyset,x'_j)\Big) \quad (4)$$

DÏ£¡FINITION 6. — *Une fonction de mapping σ est dite optimale si et seulement si*

$$\forall x,x' \in \omega d^{\sigma}_{F(\omega)}(x,x') = \min_{\sigma' \in Mappings(x,x')} d^{\sigma'}_{F(\omega)}(x,x')$$

PROPOSITION 7. — *Avec σ une fonction de mapping optimale, $d^{\sigma}_{F(\omega)}$ est une fonction de dissimilarité normalisée sur $F(\omega)$.* [2]

2.2.3. Mappings

La fonction de mapping σ détermine donc comment les attributs seront comparés entre eux. On voudra alors apparier les attributs les plus similaires. Pour ce faire plusieurs options sont possibles.

2.2.3.1. Séparation des attributs par type

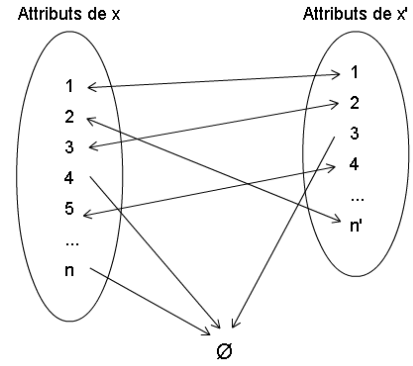
Une part non négligeable des méta-attributs des attributs n'est calculable que sur un type d'attribut particulier, entraînant de nombreuses valeurs manquantes dans la comparaison d'attributs de types différents. Ceci justifie selon (Smid, 2016) de considérer séparément les attributs de type différent dans le problème d'appariement. Ne considérant que les types numérique et nominal, cette séparation donne lieu à des mappings du type décrit en figure 3c, qui seront qualifiés de «Split ». Si au contraire on apparie simultanément les attributs numériques et nominaux, on obtient des mappings injectifs du plus grand ensemble d'attributs vers le plus petit (voir figure 3b).

2.2.3.2. Méthode de minimisation

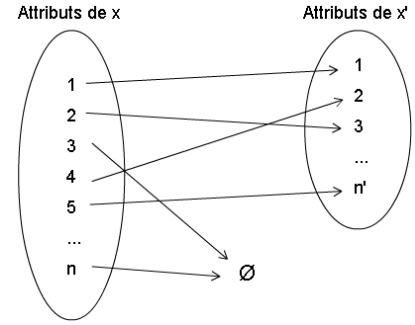
Une fonction de mapping optimale apparie les attributs de manière à minimiser la dissimilarité résultante $d^{\sigma}_{F(\omega)}(x,x')$. On peut donc considérer la méthode de recherche de ce minimum comme part intégrante de la fonction de mapping. Une méthode de recherche exacte est proposée dans (Smid, 2016), selon l'algorithme de Kuhn-Munkres (ou *algorithme hongrois*) (Kuhn, 1955). Ce dernier adresse le problème d'affectation, usuellement représenté de la façon suivante : *Soit x équipes et y tâches, $x \geq y$, et une matrice $x \times y$ de réels positifs, contenant le temps nécessaire à chaque équipe pour réaliser chaque tâche. On souhaite affecter chaque tâche à une équipe afin de minimiser le temps total de réalisation, c'est-à-dire la somme des temps pris pour chaque tâche.*

Prenons donc $x, x' \in \omega$ possédant respectivement n et n' attributs, avec $n \geq n'$. On identifie alors aux équipes les attributs de x et aux tâches ceux de x'. On doit ainsi

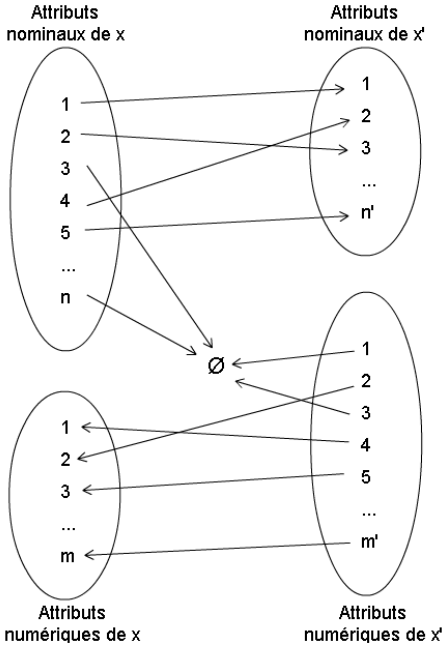
2. Voir les *Ressources* en section 5 pour la preuve.



(a) Exemple de fonction de Mapping



(b) Mapping Mix sans séparation des attributs par types



(c) Mapping Split avec séparation des attributs par type

Figure 3. Fonctions de Mapping

calculer la matrice 1 des dissimilarités entre les attributs de x et x', ce qui représente une complexité en $\mathcal{O}(n^2\ c_F)$ avec c_F la complexité de δ_F.

Tableau 1. Matrice des dissimilarités entre les attributs de x et x'

	1	...	n'
1	$\delta_F(x_1, x'_1)$	.	$\delta_F(x_1, x'_{n'})$
...	.	.	.
...	.	.	.
...	.	.	.
n	$\delta_F(x_n, x'_1)$	.	$\delta_F(x_n, x'_{n'})$

L'algorithme de Kuhn-Munkres plus tard révisé par Edmonds et Karp, et indépendament Tomizawa (Frank, 2005), fournit alors en temps polynomial ($\mathcal{O}(n^3)$) une solution optimale à ce problème d'affectation. Cette solution prend la forme d'une affectation des n' attributs de x' à des attributs distincts de x (voir tableau 2). En associant alors les potentiels attributs surnuméraires (non affectés) de x à $\emptyset$, on obtient le mapping recherché.

Tableau 2. Forme d'une solution au problème d'affectation

	1	...	n'
1	$\delta_F(x_1, x'_1)$	.	$\delta_F(x_1, x'_{n'})$
...	.	.	.
...	.	.	.
...	.	.	.
n	$\delta_F(x_n, x'_1)$	.	$\delta_F(x_n, x'_{n'})$

L'application de cette méthode exacte de minimisation pouvant se révéler très couteuse, on considérera à des fins de comparaison l'utilisation d'une méthode de minimisation naïve de complexité moindre détaillée en Algorithme 1. Cette méthode en $\mathcal{O}(\frac{n^2}{2} c_F)$ sera en général non optimale. Or, si l'on construit $d^{\sigma}_{F(\omega)}$ sur une fonction de mapping non optimale, on perd la symétrie et l'indiscernabilité des identiques, et donc la garantie d'avoir une fonction de dissimilarité normalisée.

Algorithme 1 : Méthode de minimisation naïve : Greedy

Att_x $\{1...n\}$
pour tous les *Attributs* j *de* x' **faire**
  pour tous les *Attributs* i *de* x *dans* Att_x **faire**
    Calculer $\delta_F(x_i, x'_j)$
  k i tel que $\delta_F(x_i, x'_j) = \min_{a \in Att_x} \delta_F(x_a, x'_j)$
  Associer x_k à x'_j
  Retirer k de Att_x
pour tous les *Attributs* i *de* x *restant dans* Att_x **faire**
  Associer x_i à $\emptyset$

Ces méthodes de minimisation fonctionnent indépendamment de la séparation des attributs par type. En effet, si l'on sépare les attributs numériques et nominaux, on résout simplement deux problèmes d'affectation plus simples.

2.2.3.3. Récapitulatif

La complexité des méthodes de minimisation exposées ci-avant dépend de celle de la fonction δ_F. Cette fonction comprend un normalisation par la borne supérieure, donc potentiellement couteuse sur de grands ensembles si cette borne doit être calculée. On s'affranchit de ce problème à l'implémentation, en maintenant un registre de maxima des ensembles à normaliser. Ceci permet de calculer la dissimilarité entre deux attributs δ_F en $\mathcal{O}(card(F))$, complexité de l'ordre du nombre de méta-attributs des attributs. On récapitule les différentes fonction de mapping dans le tableau 3. On notera que bien que les ordres de complexité soient équivalents, les méthodes Split seront en général de complexité inférieure, et dans le pire cas ou tous les attributs sont de même type, équivalentes aux méthodes Mix.

Tableau 3. Fonctions de mapping considérées

	Complexité	Description
GreedyMix	$\mathcal{O}(\frac{card(F)}{2}n^2)$	Méthode de minimisation naïve appliquée aux attributs numériques et nominaux.
GreedySplit	$\mathcal{O}(\frac{card(F)}{2}n^2)$	Méthode de minimisation naïve appliquée séparément aux attributs numériques et nominaux.
ExactMix	$\mathcal{O}(n^3)$	Algorithme Hongrois appliqué simultanément aux attributs numériques et nominaux.
ExactSplit	$\mathcal{O}(n^3)$	Algorithme Hongrois appliqué séparément aux attributs numériques et nominaux.

2.2.4. Composition

Par le biais des équations (2) et (4), on peut donc proposer $\forall x, y \in \omega$ la **dissimilarité normalisée par la borne supérieure** sur ω selon $d^{ubr}_{G(\omega)}$ et $d^{\sigma}_{F(\omega)}$ telle que :

$$d^{ubr}_{\omega}(x,y) = \frac{d^{ubr}_{G(\omega)}(x,y)}{\max\limits_{(x',y')\in\omega^2}(d^{ubr}_{G(\omega)}(x',y'))} + \frac{d^{\sigma}_{F(\omega)}(x,y)}{\max\limits_{(x',y')\in\omega^2}(d^{\sigma}_{F(\omega)}(x',y'))} \quad (5)$$

D'après la Proposition 1, d^{ubr}_{ω} est bien une fonction de dissimilarité normalisée sur ω.

3. Expériences comparatives

Au vu des résultats encourageants obtenus dans (Raynaut *et al.*, 2016) et (Smid, 2016), on peut étudier plus en détail l'impact en termes de performances des différents composants de la dissimilarité. Nous avons donc implémenté des variantes de ces

différents composants, et développé un cadre expérimental approprié à leur évaluation. Cette section présente le méta-problème sur lequel sont évaluées les dissimilarités ainsi que les dissimilarités étudiées. Nous détaillons ensuite les expériences effectuées et leurs résultats.

3.1. Forme du méta-problème

Les résultats de (Raynaut *et al.*, 2016) suggèrent que les dissimilarités prenant en compte les méta-attributs des attributs seraient particulièrement appropriées pour caractériser la performance d'algorithmes d'apprentissage sur des jeux de données. On s'attache donc ici à construire un méta-problème exploitant cette caractéristique.

Dans (Raynaut *et al.*, 2016), les dissimilarités étaient évaluées sur un problème de méta-classification, où la *classe* d'un jeu de données était l'algorithme qui y obtenait les meilleures performances. Il s'agit de l'une des premières approches de méta-apprentissage, ayant l'avantage d'être très simple, mais de nombreuses méthodes plus performantes ont depuis été développées (Giraud-Carrier *et al.*, 2004). Par exemple, dans (Brazdil *et al.*, 1994), le méta-problème est divisé en autant de sous-problèmes que de classifieurs, et l'on apprend alors pour chacun un modèle d'applicabilité. Dans (Kalousis, Hilario, 2001b), on apprend plutôt un modèle pour chaque paire de classifieurs, prédisant sur quels jeux de données chacun dominera. Cette décomposition du méta-problème par paire de classifieurs est reprise dans (Sun, Pfahringer, 2013), où des ensembles de règles sont appris pour comparer les performances des classifieurs dans chaque paire. D'autres travaux brisent le cadre traditionnel du problème de méta-apprentissage, tel (Leite *et al.*, 2012), qui au lieu d'utiliser des ensembles de méta-données décrivant la performance de divers algorithmes sur des jeux de données, propose une stratégie de recherche d'algorithmes performants minimisant le nombre de tests à réaliser. De même, (Sun *et al.*, 2012) considère l'optimisation des hyperparamètres en plus de la sélection d'algorithmes, et utilise des techniques d'optimisation pour chercher des solutions performantes.

Dans le cas présent, on peut définir un méta-problème s'appuyant sur l'abondante méta-donnée d'OpenML (Vanschoren *et al.*, 2012) (importante base collaborative d'expériences d'apprentissages), qui devrait idéalement refléter les points forts supposés de la dissimilarité. Afin de caractériser la performance individuelle des algorithmes, on peut reprendre une division du méta-problème par algorithme (Brazdil *et al.*, 1994). En revanche plutôt que de se limiter à prédire si les algorithmes sont applicables ou non, la dissimilarité permet de manière très intuitive de modéliser directement leur performance. En effet, un simple algorithme de type *k plus proches voisins* permet d'agréger la performance d'un algorithme sur les k jeux de données les plus proches (selon la dissimilarité). Le pseudocode de l'Algorithme 2 décrit la structure du méta-problème retenue.

En plus de la dissimilarité elle-même, certains éléments de ce procédé peuvent varier, et impacteront potentiellement les résultats d'expériences. Il conviendra donc de considérer différentes valeurs pour ces paramètres, pour autoriser un maximum de

Algorithme 2 : Forme d'une solution au méta-problème

Data : – Un meta-dataset $\mathcal{M}$ décrivant la performance de x classifieurs sur y jeux de données
– Un nouveau jeu de données D

Result : La recommandation d'un ensemble de n algorithmes $c_1...c_n$ de $\mathcal{M}$ supposés capables de bonnes performances sur D, et leur intérêt relatif attendu $\alpha_1...\alpha_n$

$Voisins \leftarrow k$ jeux de données du meta-dataset les plus proches de D selon la dissimilarité considérée

foreach *Classifieur c de* $\mathcal{M}$ **do**
  Estimer la performance inconnue de c sur D selon la performance connue de c sur les $Voisins$ de D

Ordonner les classifieurs du meta-dataset selon l'estimation de leur performance sur D

Recommander les n meilleurs pondérés selon leur performance estimée

généralité aux résultats d'expérience. Le nombre k de voisins considérés prendra des valeurs communes pour des ensembles de l'ordre de la centaine :

$$k \in \{3, 5, 10\}$$

L'estimation de la performance d'un classifieur c sur D selon sa performance sur les voisins de D pourra soit être une simple moyenne des performances de c sur les voisins de D, soit être pondérée par la dissimilarité :

$$\text{perf}_{mean}(c, D) = \frac{1}{k} \sum_{V \in Voisins} \text{perf}(c, V)$$

$$\text{perf}_{weighted}(c, D) = \frac{\sum_{V \in Voisins} d_{\omega}^{ubr}(D, V) * perf(c, V)}{\sum_{V \in Voisins} d_{\omega}^{ubr}(D, V)}$$

De plus, si l'on accepte ainsi en sortie un *ensemble* d'algorithmes recommandés, on doit définir un critère de performance capable d'évaluer des solutions au méta-problème produisant de tels ensembles. On généralise ainsi le critère introduit dans (Raynaut *et al.*, 2016) :

Dï£¡FINITION 8. — *Soit une solution $\mathcal{S}$ au méta-problème $(\mathcal{M}, D)$ recommandant les classifieurs $c_1...c_n$ avec un poids relatif $\alpha_1...\alpha_n$. Soient $p_1...p_n$ les performances réelles respectives de $c_1...c_n$ sur D. Soient alors **best** la meilleure performance obtenue par un classifieur de $\mathcal{M}$ sur le jeu de données D, et **def** la performance du classifieur par défaut (prédisant la classe majoritaire) sur D. On définit tout d'abord la performance d'une recommandation c_i :*

$$perf(c_i) = \max(-1, 1 - \frac{|best - p_i|}{|best - def|})$$

Ce critère, illustré en figure 4, reprend celui de (Raynaut et al.*, 2016), atteignant son maximum de* 1 *quand le classifieur recommandé présente une valeur de précison maximale, et* 0 *quand il présente la même précision que le classifieur par défaut. On limite cependant ce critère en* -1 *car il fait peu de sens de discriminer entre des recommandations inutiles. On définit alors la performance de notre solution* $\mathcal{S}$ *au méta-problème* $(\mathcal{M}, D)$ *:*

$$perf(\mathcal{S}) = \frac{\sum_{i \in \{1...n\}} \alpha_i * perf(c_i)}{\sum_{i \in \{1...n\}} \alpha_i}$$

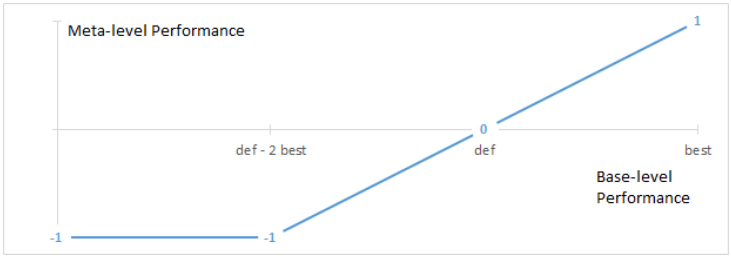


Figure 4. Performance d'une recommandation c_i

Ce nouveau critère ne peut prendre de valeurs extrêmes que plus difficilement, requérant pour ce faire des recommandations unanimement extrêmes. Ceci devrait limiter la variance des résultats, mais pour étudier plus précisément l'impact en terme de performances du nombre de recommandations on fera varier le nombre d'algorithmes recommandés en sortie :

$$n \in \{1, 3, 5\}$$

3.2. Ensembles de méta-attributs

Les méta-attributs retenus dans cette expérience sont en partie repris des travaux de (Smid, 2016) et reprennent des mesures classiques additionnées de diverses variations plus particulières. On les divise en différents ensembles selon leur provenance afin d'évaluer l'impact de l'utilisation de différents méta-attributs sans trop augmenter la taille de l'expérience. On définit pour chaque méta-attribut une dissimilarité bornée sur son ensemble de valeurs adjoint de $\emptyset$, comme une distance de Manhattan sur son ensemble de valeurs et une distance à l'absence de valeur particulière sur $\emptyset$. Soit donc pour un méta-attribut a sa dissimilarité associée $d_a : (a(\omega) \cup \emptyset)^2 \mapsto \mathbb{R}^+$ telle que :

$$d_a(x,y) = \begin{cases} |x-y| & si\ \ (x,y) \in a(\omega)^2 \\ \delta_a^{\emptyset}(x) & si\ \ x \in a(\omega)\ \ et\ \ y = \emptyset \\ \delta_a^{\emptyset}(y) & si\ \ y \in a(\omega)\ \ et\ \ x = \emptyset \\ \delta_a^{\emptyset}(\emptyset) & si\ \ x = y = \emptyset \end{cases}$$

On pourra se référer aux *Ressources* en section 5 pour les listes complètes des méta-attributs retenus et leurs distances à l'absence de valeur $\delta^{\emptyset}$ associées, omises ici pour des raisons de volume. Cette distance à l'absence de valeur sera en général une distance de Manhattan à la valeur du méta-attribut considéré sur un attribut hypothétique vide ou uniforme (n'ayant qu'une seule valeur). Un tel attribut n'apporte en effet aucune information et est en cela identifiable à l'absence d'attribut. Pour certains méta-attributs, ce raisonnement n'est pas possible ou ne fait aucun sens (par exemple, comparer la moyenne d'un attribut avec celle d'un attribut vide est absurde), et l'on y considérera $\delta^{\emptyset}(x) = 0$. On associe une distance à l'absence de valeur à la fois aux méta-attributs généraux des jeux de données et à ceux des attributs, car elle est nécessaire à la définition de la dissimilarité sur les attributs, et assure une certaine robustesse aux valeurs manquantes de méta-attributs généraux.

3.2.1. Méta-attributs généraux des jeux de données

Les méta-attributs généraux des jeux de données consistent en des propriétés simples des jeux de données, un descriptif global des attributs numériques, un descriptif global des attributs nominaux, et en la performance de *landmarkers* évalués selon plusieurs critères (voir les *Ressources* en section 5 pour des listes complètes). Les *landmarkers* sont des algorithmes d'apprentissage simples, dont l'usage a été introduit dans (Pfahringer *et al.*, 2000), que l'on applique au jeu de données considéré, afin d'y évaluer leur performance. Ceux utilisés ici proviennent de l'API Weka (Hall *et al.*, 2009) et rassemblent différentes techniques classiques d'apprentissage. Les critères de performance retenus sont l'aire sous la courbe ROC (*Receiver Operating Characteristic*), le taux d'erreur et le coefficient Kappa de Cohen (Cohen, 1968), qui, parmi les critères communément utilisés, capturent des aspects de la performance conceptuellement assez différents. On forme alors trois ensembles de méta-attributs généraux comprenant tous nos trois premiers ensembles de base (propriétés simples des jeux de données, descriptif global des attributs numériques, descriptif global des attributs nominaux) et différents ensembles de *landmarkers* :

DMFg_min : Aucun *landmarker*.
DMFg_red : AUC (aire sous la courbe ROC) des *landmarkers*.
DMFg_full : Tous les *landmarkers*.

3.2.2. Méta-attributs des attributs

Les différents méta-attributs retenus pour les attributs individuels de jeux de données consistent en des propriétés simples communes à tous types d'attributs, des propriétés exclusives aux attributs numériques, des propriétés exclusives aux attributs nominaux, et des versions normalisées de certaines des propriétés précédentes (voir les *Ressources* en section 5 pour des listes complètes). Cette normalisation est proposée

dans (Smid, 2016), suite au constat que les distributions de certains méta-attributs sur un ensemble de jeux de données courants peuvent se révéler peu informatives. En effet, certains méta-attributs sont fortement corrélés à la taille (nombre d'instances) du jeu de données, comme par exemple le *nombre* de valeurs manquantes. La solution proposée et d'ajouter une nouvelle version de ces méta-attributs, normalisée par le nombre d'instances. On forme alors deux ensembles de méta-attributs des attributs :

DMFf_base : Pas de méta-attributs normalisés.
DMFf_full : Tous les méta-attributs normalisés.

3.3. Fonctions de dissimilarité

On définit ici les différentes dissimilarités à comparer. Les éléments nécessaires à la définition d'une dissimilarité particulière sont, d'une part des ensembles de méta-attributs généraux et méta-attributs des attributs, tels que présentés dans la section précédente, et d'autre part des fonctions de dissimilarité sur ces méta-attributs géné-raux et méta-attributs des attributs. On présente donc les fonctions considérées, avant de lister les dissimilarités ainsi formées pour comparaison.

3.3.1. Dissimilarités sur les méta-attributs généraux

Pour fournir une dissimilarité sur les méta-attributs généraux, on compare le candidat présenté en définition 3, la dissimilarité normalisée par la borne supérieure d_G^{ubr}, à des distances classiques. On considère un simple panel constitué des distances euclidienne (norme 2), de Manhattan (norme 1), et de Tchebychev (norme infinie) :

dissimG Dissimilarité normalisée par la borne supérieure
distEucl Distance Euclidienne
distMan Distance de Manhattan
distTcheb Distance de Tchebychev

3.3.2. Dissimilarités sur les attributs

Une dissimilarité sur les méta-attributs des attributs a été définie dans l'équation 3, se basant sur les différentes méthodes d'appariement des attributs décrites en section 2.2.3. Les dissimilarités construites selon ces différentes méthodes d'appariement peuvent alors être comparées entre elles.

D'autre part, des techniques existent dans le domaine du test statistique pour comparer directement des distributions. En identifiant un attribut de jeu de données à une distribution, on pourrait utiliser de telles techniques pour construire une dissimilarité entre attributs. On considère donc le test de Kolmogorov-Smirnov, permettant de tester la significativité des différences entre deux échantillons de données. Ce dernier à l'avantage d'être non-paramétrique (aucun pré-requis sur les distributions comparées), ce qui nous permet de l'appliquer directement à tout attribut numérique. Afin de pouvoir l'appliquer également à des attributs nominaux, on considère une simple association d'index entiers uniques aux catégories. On peut alors définir une nouvelle fonction de dissimilarité δ_{KS} sur les attributs de jeux de données comme la statistique

résultante d'un test de Kolmogorov-Smirnov pour l'hypothèse nulle selon laquelle deux attributs proviennent d'une même distribution :

Dï£¡FINITION 9. — *Soient x et y deux jeux de données de ω. On note x_i le i^{th} attribut de x, et ω_a l'ensemble des attributs des jeux de données de ω. On note $KS(H_0)$ la statistique résultante d'un test de Kolmogorov-Smirnov pour l'hypothèse nulle H_0. On définit alors $\delta_{KS} : \omega_a^2 \mapsto \mathbb{R}^+$ telle que :*

$$\delta_{KS}(x_i, y_j) = KS(x_i \textit{ et } y_j \textit{ sont issus de la même distribution})$$

PROPOSITION 10. — *δ_{KS} est une fonction de dissimilarité normalisée.* [3]

Afin de construire une fonction de dissimilarité sur les attributs à partir de δ_{KS}, on reprend l'équation (4). Soient donc $x, x' \in \omega$ ayant respectivement n et n' attributs. Étant donnée une fonction de mapping σ, on peut définir $d^{\sigma}_{KS(\omega)} : \omega^2 \mapsto \mathbb{R}^+$ telle que :

$$d^{\sigma}_{KS(\omega)}(x, x') = \frac{1}{max(n, n')}\Big(\sum_{\sigma_1(i)=j}^{i,j} \delta_{KS}(x_i, x'_j) + \sum_{\sigma_1(i)=\emptyset}^{i} \delta_{KS}(x_i, \emptyset) + \sum_{\sigma_2(j)=\emptyset}^{j} \delta_{KS}(\emptyset, x'_j)\Big) \tag{6}$$

δ_{KS} étant bien une fonction de dissimilarité normalisée, la proposition 7 tient toujours et assure que $d^{\sigma}_{KS(\omega)}$ est une fonction de dissimilarité normalisée sur les attributs des jeux de données de ω. δ_{KS} ne nécessitant pas à proprement parler de méta-attributs des attributs, on notera **DMFf_dist** l'ensemble des distributions des attributs.

On compare donc deux dissimilarités sur les méta-attributs des attributs, d^{σ}_{F} et d^{σ}_{KS}, utilisant les différents *mappings* σ décrits dans le tableau 3 : **greedyMix**, **exactMix**, **greedySplit** et **exactSplit**. On utilise bien sûr d^{σ}_{KS} sur **DMFf_dist** et d^{σ}_{F} sur **DMFf_base** et **DMFf_full**.

3.4. Cadre d'expérimentation

3.4.1. Meta-Dataset

Afin d'instancier le méta-problème décrit en section 3.1, on doit construire un *méta-dataset* décrivant la performance d'un ensemble de classifieurs sur un ensemble de jeux de données. On raffine pour cela la procédure employée dans (Raynaut *et al.*, 2016), pour construire ce *méta-dataset* depuis les données d'OpenML.

On se limite tout d'abord à quatre critères de performances bien distincts. En effet, les résultats de (Raynaut *et al.*, 2016) ont montré que nombre de critères de performance menaient à des résultats très corrélés. Réduire le nombre de critères permet

3. Voir les *Ressources* en section 5 pour la preuve.

donc de limiter la complexité de l'expérience sans trop impacter la généralité des résultats. Les critères retenus sont présentés dans le tableau 4.

Tableau 4. Critères de performance retenus

Critère	Description
area_under_roc_curve	The area under the ROC curve (AUROC), calculated using the Mann-Whitney U-test.
predictive_accuracy	The Predictive Accuracy is the percentage of instances that are classified correctly.
kappa	Cohen's kappa coefficient is a statistical measure of agreement for qualitative (categorical) items: it measures the agreement of prediction with the true class.
kb_relative_information_score	The Kononenko and Bratko Information score, divided by the prior entropy of the class distribution, measures the information produced by the model.

Pour trouver des ensembles de classifieurs et de jeux de données tels que chaque classifieur ait été évalué sur chaque jeu de données selon nos quatre critères choisis, on utilise une technique de recherche de bi-clique maximale (Uno *et al.*, 2004) pour trouver les plus grands ensembles de jeux de données et de classifieurs tels que chaque élément des deux ensembles a été évalué en conjonction avec tous les éléments de l'autre ensemble. Dans cette itération, on se limite à des jeux de données d'au plus cent attributs, afin là-encore de limiter la complexité de l'expérience (dont un facteur de complexité déterminant est celui des méthodes d'appariement des attributs). Les ensembles ainsi produits, de respectivement 48 classifieurs et 395 jeux de données, ainsi que le *méta-dataset* complet au format *ARFF*, sont mis à disposition (voir les *Ressources* en section 5).

3.4.2. Baseline

Les dissimilarités à comparer au méta-niveau (voir les *Ressources* en section 5 pour une liste complète) seront évaluées selon le protocole décrit par l'algorithme 2. Une première partie de notre *baseline* est constituée des distances classiques qui y sont présentées, mais une comparaison à des méthodes d'apprentissage traditionnelles reste désirable. On introduit donc une légère variante du méta-problème permettant d'évaluer des algorithmes d'apprentissage traditionnels dans des circonstances semblables, venant ainsi les ajouter à notre *baseline*. Ce protocole, présenté dans l'algorithme 3, permettra de comparer nos approches utilisant des dissimilarités à des algorithmes d'apprentissage classiques, sélectionnés pour représenter des biais aussi divers que possible. Voir les *Ressources* en section 5 pour une liste détaillée.

3.4.3. Exécutions

On évalue donc nos algorithmes d'apprentissage et dissimilarités sur le meta-dataset, respectivement selon les algorithmes 3 et 2. On explore ainsi l'espace formé sur les dimensions décrites dans le tableau 5.

Algorithme 3 : Méta-problème résolu par un algorithme d'apprentissage traditionnel $\mathcal{A}$

Data : – Un meta-dataset $\mathcal{M}$ décrivant la performance de x classifieurs sur y jeux de données
– Un nouveau jeu de données D

Result : La recommandation d'un ensemble de n algorithmes $c_1...c_n$ de $\mathcal{M}$ supposés capables de bonnes performances sur D, et leur intérêt relatif attendu $\alpha_1...\alpha_n$

foreach *Classifieur* c *de* $\mathcal{M}$ **do**
  Construire avec $\mathcal{A}$ un modèle de la performance de c à partir de $\mathcal{M}$
  Prédire la performance de c sur D selon ce modèle.
Ordonner les classifieurs du meta-dataset selon l'estimation de leur performance sur D
Recommander les n meilleurs pondérés selon leur performance estimée

Les dissimilarités apparaissent plus nombreuses que prévu car leur nombre est multiplié par les dimensions présentées dans la définition du méta-problème. En effet, le nombre k de voisins considérés dans l'algorithme 2 et la méthode $nnDist$ d'estimation de performance d'un classifieur à partir de celle de ses voisins, sont des dimensions internes des dissimilarités. Les dimensions de l'espace des dissimilarités sont présentées dans le tableau 6.

L'expérience complète nécessite donc 33 180 exécutions de l'algorithme 3 et 1 279 800 exécutions de l'algorithme 2 pour évaluer la performance au méta-niveau en chaque point de l'espace. Ceci nécessite un important degré de parallélisme pour être calculé en un temps raisonnable, ici obtenu en générant dynamiquement les exécutions (toutes indépendantes) et en les déléguant à un répartiteur de tâches SLURM (Yoo *et al*., 2003) gérant les 640 nœuds du cluster *OSIRIM* (voir osirim.irit.fr). De plus, un pré-calcul des dissimilarités a été effectué afin de minimiser la redondance entre les exécutions. Ce pré-calcul a de plus permis de mesurer le temps de calcul exact des différentes dissimilarités, présenté en figure 5.

Tableau 5. Dimensions de l'expérience

Dimension	Détails	Taille
Algorithme au méta-niveau	Traditionnels	7
	Dissimilarités	270
Critère de performance de base	Tableau 4	4
Jeu de données (instance de base)	Voir *Ressources*	395
Nombre n d'algorithmes de base recommandés	$\{1, 3, 5\}$	3

Tableau 6. Dimensions de l'espace des dissimilarités

Dimension	Détails	Taille
Méta-attributs généraux	Section 3.2.1	3
Dissimilarité sur les méta-attributs généraux	Section 3.3.1	4
Méta-attributs des attributs	Section 3.2.2	3
Dissimilarité sur les attributs	Tableau Section 3.3.2	4
Nombre k de voisins considérés	$\{3, 5, 10\}$	3
Méthode $nnDist$ d'estimation de performance d'un classifieur	{ *mean, weighted* }	2

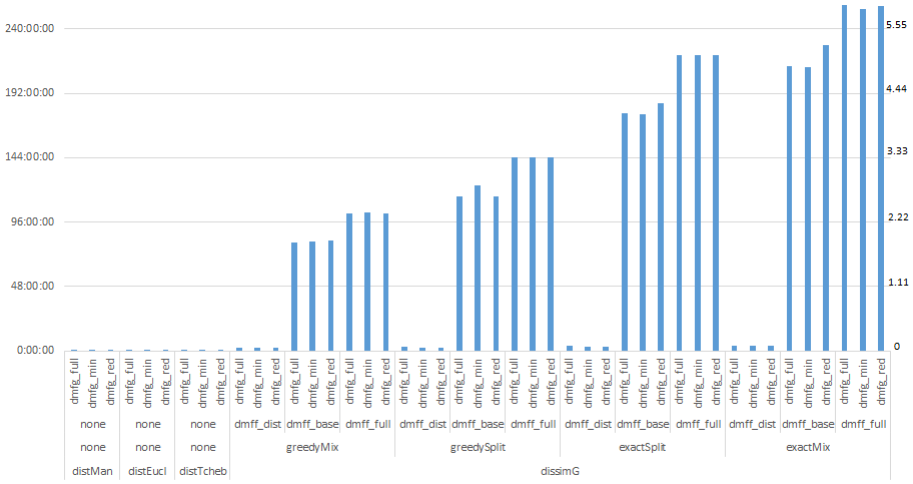


Figure 5. Temps de calcul des dissimilarités. L'échelle en heures à gauche présente le temps total de calcul entre chaque paire de jeux de données, et l'échelle en secondes à droite présente la durée moyenne d'une évaluation de la dissimilarité

Ces temps d'exécution sont globalement compatibles avec les complexités attendues, montrant une forte dépendance envers la méthode d'appariement des attributs, et le nombre de méta-attributs des attributs. La dissimilarité sur les attributs basée sur le test de Kolmogorov présente des temps d'exécution bien moindres (de 97 à 99%) que celle basée sur les méta-attributs, pour les ensembles étudiés ici (37 à 49 méta-attributs).

3.5. Analyse dimensionnelle des résultats

Les 1 312 980 exécutions détaillées dans la section précédente résultent en autant de valeurs de performance au méta-niveau. Ces résultats sont stockés dans une base de

données SQL, dont les mécanismes sont propices à l'analyse dimensionnelle. On étudiera ainsi l'influence individuelle des différentes dimensions sur les performances au méta-niveau pour tenter d'y déceler des tendances, qui devront ensuite être soumises à des tests d'hypothèse statistique pour validation. On utilisera en particulier le test de Friedman sous l'hypothèse nulle d'identité des distributions pour valider l'existence d'une tendance, et le test de Nemenyi pour en établir le sens. Cette procédure est détaillée ici sur l'exemple du nombre k de voisins considérés par l'algorithme 2, tandis que les résultats obtenus sont présentés et interprétés en section suivante [4].

3.5.1. Processus d'analyse

L'algorithme 2 estime la performance des classifieurs sur un nouveau jeu de données D selon leur performance sur les k plus proche voisins de D, au sens de la dissimilarité évaluée. Ce nombre k prend ici des valeurs communes pour des ensembles de l'ordre de la centaine (Batista, Silva, 2009) : $k \in \{3, 5, 10\}$. Pour étudier l'impact de ce facteur k, on observe le comportement des différentes dissimilarités pour chaque valeur de k, en moyennant selon les autres dimensions. Cette performance moyenne au méta-niveau est présentée en figure 6.

On rappelle que ces valeurs de performance moyenne sont un pourcentage du maximum connu : par exemple, la dissimilarité *dissimG - DMFg_full - greedySplit - DMFf_full* affiche une performance moyenne de 0.87 pour $k = 10$, ce qui signifie que les exécutions de l'algorithme 2 sur cette dissimilarité avec $k = 10$ ont permis de trouver des classifieurs en moyenne $87\ \%$ aussi performant que le meilleur. En observant la figure 6, on peut conjecturer plusieurs tendances :

1) La performance au méta-niveau pour $k = 3$ semble *souvent* inférieure à celle obtenue pour $k = 5$ et $k = 10$.

2) Pour les dissimilarités utilisant *DMFf_full*, les performances au méta-niveau apparaissent toujours croissantes de $k = 3$ à $k = 5$ puis $k = 10$.

Afin de contrôler le risque que ces tendances ne reflètent pas de réelles différences entre nos distributions, on fait appel aux tests d'hypothèse statistique de Friedman et Nemenyi. Le test de Friedman est un test non paramétrique, ne posant aucune condition sur la forme des distributions sous-jacentes, ce qui est nécessaire dans ce contexte multidimensionnel où aucune distribution n'est connue. Il permet de comparer des échantillons de valeurs dans le but d'assurer l'improbabilité de l'hypothèse nulle $H_0 = \{$ Les différents échantillons sont tirés de la même distribution $\}$. La *p-value* retournée par le test de Friedman (apparaissant dans les figures type 7a) mesure ainsi la probabilité de l'observation faîte sous H_0. Ici, cela signifie que si H_0 est vrai ($\leftrightarrow$ si le facteur k n'a pas de réelle influence sur la performance), alors la probabilité d'observer les distributions en figure 6 était de $9,2496 * 10^{-13}$ (*p-value* en figure 7a). Ceci nous assure que H_0 est hautement improbable : le facteur k a bien une influence

4. Voir les *Ressources* en section 5 pour les détails de l'analyse complète de chaque facteur.

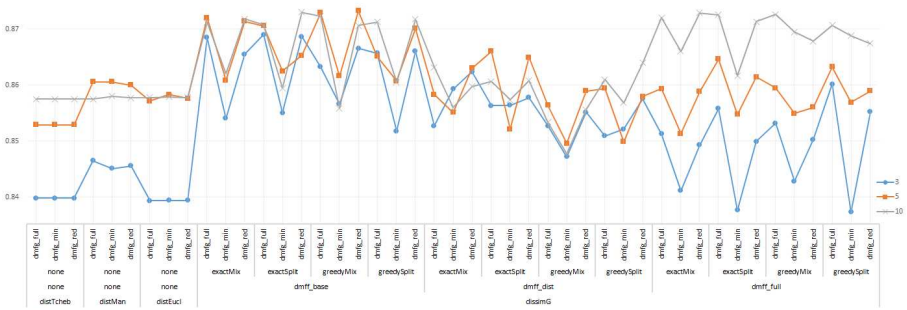
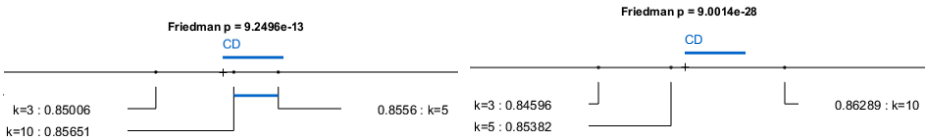


Figure 6. Moyenne des performances au méta-niveau selon le nombre k de voisins considérés

sur la performance, mais cela ne suffit pas à valider l'existence des tendances relevées plus haut.

Pour ce faire, on fait appel au test *post-hoc* de Nemenyi. Ce dernier permet de comparer les échantillons deux à deux, déterminant lesquels sont significativement différents, tout en controllant le risque global d'erreur type 1. En effet, comparer de nombreux échantillons rend exponentiel le risque de commettre au moins une erreur de type 1 (de valider une différence infondée). Le test de Nemenyi permet de maitriser ce risque quel que soit le nombre d'échantillons. Dans nos expériences, on contraindra ce risque à la valeur courante de $0,05$, ce qui signifie que chaque test de Nemenyi effectué a un risque d'au plus $5\ \%$ de discriminer deux échantillons qui n'étaient pas réellement discernables. Le test résulte ainsi en une *différence critique* CD, représentant la différence nécessaire entre le *rang moyen* de deux échantillons (statistique produite par le test de Friedman) pour pouvoir les considérer significativement différents. Ceci est représenté en figure 7a, où l'on classe les différents échantillons par rang moyen. Ceux trop proches pour pouvoir être considérés significativement différents sont liés entre eux. Ici, cela signifie que l'échantillon $k = 3$ est significativement moins bon que les échantillons $k = 10$ et $k = 5$, mais que ces derniers ne sont pas suffisamment éloignés pour être jugés significativement différents (sans courir un risque d'erreur de plus de $5\ \%$). La valeur indiquée à côté de l'identifiant de l'échantillon est sa performance moyenne, qui permet de constater les écarts de moyenne parfois très faibles entre échantillons jugés différents. Les résultats du test de Nemenyi en figure 7a valident donc notre première tendance : choisir $k = 3$ mène à des performances significativement inférieures. Pour valider la seconde, on répète les tests de Friedman et Nemenyi en se limitant cette fois aux dissimilarités utilisant *DMFf_full*. Les résultats, présentés en figure 7b, montrent bien que $k = 10$ y est significativement meilleur que $k = 5$, lui-même significativement meilleur que $k = 3$.

Ces différents résultats nous permettent d'écarter $k = 3$, et pointe vers l'utilisation de $k = 10$, en particulier en conjonction avec *DMFf_full*. Ceci pourrait indiquer



(a) Résultats du test de Friedman avec *post-hoc* de Nemenyi entre les échantillons représentant les différentes valeurs de k

(b) Idem mais en se restreignant aux dissimilarités utilisant *DMFf_full*

Figure 7. Résultats des tests de Friedman

la nécessité de considérer davantage de voisins si l'on souhaite exploiter de grands ensembles de méta-attributs des attributs.

Cette méthodologie de test, par étude des distributions puis validation par Friedman et Nemenyi, guide ainsi notre étude des résultats, et est reproduite sur nos différentes dimensions pour évaluer l'impact de chaque facteur identifié plus tôt.

4. Discussion

4.1. Récapitulatif des résultats

On résume ici les différents résultats significatifs obtenus et leur interprétation. Afin de présenter les résultats de tests d'hypothèse de manière synthétique, on utilisera les notations $\prec$ et $\preceq$ pour *est dominé significativement* et *est dominé sans différence significative*, respectivement.

4.1.1. Facteurs secondaires

On qualifie de secondaires les facteurs pouvant influer sur la performance au méta-niveau mais ne faisant pas partie intégrante des dissimilarités. On étudie ici leur impact sur la performance au méta-niveau et leurs relations avec les différentes dissimilarités.

Nombre de voisins considérés par l'algorithme 2 :

$$3 \prec 10 \preceq 5 \text{ en général}$$
$$3 \prec 5 \prec 10 \text{ sur } \textit{DMFf\_full}$$

Ceci semble indiquer la nécessité de considérer davantage de voisins pour exploiter de grands ensembles de méta-attributs des attributs.

Méthode d'estimation de performance d'un classifieur selon celle de ses voisins :

$$\textit{Mean} \prec \textit{Weighted}$$

Le test valide la supériorité globale de la méthode **weighted** (utilisant à nouveau la dissimilarité pour pondérer la performance des plus proches voisins). Ce résultat conforte l'intérêt perçu des dissimilarités, et leur utilité pour l'estimation de performance au niveau de base.

Critère de performance de base :

$$\textit{Predictive accuracy} \prec \textit{Information score} \prec \textit{Kappa} \prec \textit{AUC}$$

L'ordonnancement des différents critères semble valide dans une portion significative des cas de test, et aucun contre-exemple significatif n'a pu être trouvé. Tous nos éléments pointent donc vers l'aire sous la courbe de ROC, dont l'utilisation comme critère de performance de base semble mener aux meilleures performances au méta-niveau, et qui est généralement reconnu comme un bon critère de performance en classification.

Nombre de recommandations :

$$5 \prec 3 \prec 1$$

Augmenter le nombre de recommandations impacte donc bien négativement les performances au méta-niveau. Ce compromis était attendu, le but étant de contrôler la variance des résultats. On observe les variances les plus basses sur le critère d'*AUC*, sans que le nombre de recommandations ne semble l'impacter. Le seul critère où la multiplication des recommandations a l'effet escompté et limite la variance, est celui de précision, mais les valeurs y restent peu intéressantes. Cette étude conforte donc le choix du critère d'aire sous la courbe de *ROC*, mais rend apparent le peu d'intérêt de la multiplication des recommandations, du point de vue des performances.

Ces études préliminaires nous ont permis de gagner une meilleure compréhension de l'impact des facteurs secondaires, et en particulier de déterminer un espace optimal, ensemble de valeurs des facteurs secondaires maximisant la performance au méta-niveau. Dans certaines des observations suivantes, on se placera dans cet espace optimal pour analyser le comportement des dissimilarités dans ce qui serait le plus proche d'un futur cas d'application réel. Comme espace optimal, on retient donc les valeurs de $k = 10$, nnDist=*weighted*, criterion=AUC, et $n = 1$. De plus, on prendra toujours la performance au méta-niveau pour $n = 1$ (sauf mention contraire explicite).

4.1.2. Facteurs primaires

On qualifie de primaires les éléments fonctionnels variables des dissimilarités.

Dissimilarité sur les méta-attributs généraux :

$$distTcheb \prec distEucl \prec distMan \prec dissimG$$

La dominance de la distance de Manhattan sur les autres distances en conforte le choix comme brique de base des dissimilarités. De plus, on valide la supériorité de la dissimilarité normalisée par la borne supérieure, ce qui confirme l'intéret de cette normalisation et de la prise en compte des valeurs de méta-attributs manquantes par dissimilarité à l'absence de valeur.

Méta-attributs généraux :

$$DMFg\_min \prec DMFg\_red \preceq DMFg\_full$$

L'utilisation de méta-attributs de type *landmarker* améliore significativement la performance, mais la multiplication des évaluations des *landmarkers* ne présente pas d'effet significatif. On peut rejoindre ici des travaux de sélection d'attributs insistant sur l'importance de la diversité au sein des attributs (Brown *et al.*, 2012). Il pourrait ainsi être intéressant d'étudier l'impact de la *diversité* des *landmarkers* et autres méta-attributs généraux choisis sur la performance au méta-niveau.

Dissimilarité sur les attributs :

$$greedyMix \prec exactSplit$$

Le coût d'exécution du *mapping exactSplit* dépassant celui du *greedyMix* d'un simple facteur 2, ces résultats pointent vers l'utilisation du *mapping* exact avec séparation des attributs par type.

Méta-attributs des attributs :

$DMFf\_none \prec DMFf\_dist \prec DMFf\_base \preceq DMFf\_full$ en général
$DMFf\_none \prec DMFf\_base \preceq DMFf\_dist \prec DMFf\_full$ sur l'espace optimal

La dominance de *DMFf_full* sur l'espace optimal confirme l'intérêt des méta-attributs normalisés, d'autant plus qu'il émerge lorsqu'on se place dans la situation de performance optimale des dissimilarités. *DMFf_none* est de plus toujours dominé, ce qui confirme ici l'intérêt d'utiliser les méta-attributs des attributs.

4.1.3. Comparatif global

La comparaison à la *baseline* forme le dernier spectre d'observations à mener. On compare ainsi directement les performances des dissimilarités et des algorithmes de la *baseline*, confirmant la supériorité des dissimilarités par rapport aux algorithmes traditionnels étudiés. Le test ne suffit malheureusement pas à établir de différence significative entre les différentes dissimilarités, mais de telles différences ont déjà pu être constatées dans les paragraphes précédents.

Ces résultats, dans l'ensemble très positifs, démontrent l'intérêt des approches proposées pour la sélection d'algorithme de classification, problème standard de méta-analyse. La proximité des biais entre différents problèmes de méta-analyse permet de supposer que ces approches par dissimilarité pourront y avoir de bons résultats, et ce moyennant très peu de modifications pour de nouveaux problèmes de sélection d'algorithme. Les performances obtenues au méta-niveau, pouvant dépasser les 0.95 sur notre ensemble de 395 jeux de données, signifient que certaines approches par dissimilarité ont permis d'identifier des algorithmes de classification en moyenne 95 % aussi performants que le meilleur sur chaque jeu de données. Le processus est couteux si pratiqué *offline* : calculer la matrice complète des dissimilarités entre jeux de données peut prendre des jours pour de grands ensembles. En revanche, dans une perspective *online*, l'ajout d'un nouveau jeu de données ne nécessite que de calculer sa dissimilarité à ceux déjà présents, au lieu de reconstruire le modèle complet. Ces approches seront donc particulièrement adaptées à des processus de sélection d'algorithmes actifs maintenant une base de cas sur lesquels construire leurs recommandations.

4.2. Conclusion

Nous avons proposé des fonctions de dissimilarité entre jeux de données présentant un ensemble de propriétés désirables, et capables d'employer des méta-attributs caractérisant des attributs particuliers de ces jeux de données. Nous avons montré qu'elles permettent de caractériser l'adéquation d'algorithmes de classification avec des jeux de données plus efficacement que des distances traditionnelles, et qu'elle peuvent être employées avec de bonnes performances dans le contexte de sélection d'algorithmes.

De nombreuses pistes d'amélioration restent cependant à explorer. Tout d'abord, ces dissimilarités permettent d'utiliser des méta-attributs caractérisant des attributs particuliers des jeux de données, mais diverses expériences (Long *et al*., 2005 ; Brown *et al*., 2012) ont montré que les propriétés d'un attribut *dans le contexte des autres attributs* sont au moins aussi importantes. Il serait alors intéressant de permettre l'utilisation de tels méta-attributs relationnels, comme la covariance ou l'information mutuelle, par la dissimilarité.

De plus, bien que divers et provenant d'approches très différentes, les méta-attributs employés dans nos expériences ne couvrent pas complètement l'état de l'art en la matière. Ces dernières années ont été riches en contributions introduisant de nouveaux méta-attributs (Peng *et al*., 2002 ; Ho, Basu, 2002 ; Ntoutsi *et al*., 2008 ; Sun, Pfahringer, 2013), dont l'utilisation pourrait révéler l'intérêt d'une approche par dissimilarité dans de nouveaux contextes.

Enfin, comme l'efficacité de l'approche par dissimilarité apparaît très dépendante du contexte (comme c'est souvent le cas en apprentissage et méta-apprentissage), il pourrait être intéressant de concevoir une méthode d'évaluation de méta-attributs considérant leurs diverses natures (globaux, liés à un attribut, relationnels...). Il serait alors possible de caractériser l'utilité des divers méta-attributs dans une variété de situations et donc d'approfondir notre connaissance du problème de méta-apprentissage.

Dans le cadre de l'assistance intelligente à l'analyse de données, un atout particulier de notre approche est qu'elle permet une caractérisation unifiée des expériences d'analyse de données. En effet, disposant d'une quelconque représentation du processus d'analyse de données et de ses résultats, il est possible de l'intégrer dans la dissimilarité, permettant ainsi la comparaison directe d'expériences complètes. Il s'agit là d'un premier pas vers de nouvelles approches d'assistance intelligente à l'analyse de données, permettant notamment l'utilisation directe d'heuristiques pour la découverte et recommandation de processus d'analyse adaptés.

5. Ressources

Se référer à https://github.com/WilliamRaynaut/Dissimilarites-entre-jeux-de-donnees pour tous les matériaux complémentaires. On trouvera tout d'abord les listes des méta-attributs, des algorithmes traditionnels de la baseline, et des classifieurs et jeux de données du meta-dataset. Le dossier contient également les preuves des différentes propositions présentées dans l'article, l'analyse complète des différents facteurs ainsi que l'intégralité des résultats produits.

Remerciements

Ce travail a été réalisé sur un financement COMUE - région Occitanie. Les expériences ont été réalisées en utilisant la plateforme OSIRIM, qui est administrée par l'IRIT et soutenue par le CNRS, la région Occitanie, le gouvernement français, et le FEDER (voir http://osirim.irit.fr/site/fr).

Bibliographie

Batista G., Silva D. F. (2009). How k-nearest neighbor parameters affect its performance. In *Argentine symposium on artificial intelligence*, p. 1–12.

Brazdil P., Gama J., Henery B. (1994). Characterizing the applicability of classification algorithms using meta-level learning. In *European conference on machine learning*, p. 83–102.

Brown G., Pocock A., Zhao M.-J., Luján M. (2012). Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The Journal of Machine Learning Research*, vol. 13, n° 1, p. 27–66.

Cohen J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, vol. 70, n° 4, p. 213.

Frank A. (2005). On kuhn's hungarian method: a tribute from hungary. *Naval Research Logistics (NRL)*, vol. 52, n° 1, p. 2–5.

Fürnkranz J., Petrak J. (2002). *Extended data characteristics*. Rapport technique. METAL consortium. (Accessed 12/11/15 at citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.97.302)

Giraud-Carrier C., Vilalta R., Brazdil P. (2004). Introduction to the special issue on meta-learning. *Machine learning*, vol. 54, n° 3, p. 187–193.

Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, vol. 11, n° 1, p. 10–18.

Ho T. K., Basu M. (2002). Complexity measures of supervised classification problems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, n° 3, p. 289–300.

Kalousis A. (2002). *Algorithm selection via meta-learning*. Thèse de doctorat non publiée, Universite de Geneve.

Kalousis A., Gama J., Hilario M. (2004). On data and algorithms: Understanding inductive performance. *Machine Learning*, vol. 54, n° 3, p. 275–312.

Kalousis A., Hilario M. (2001a). Feature selection for meta-learning. In *Proceedings of the 5th pacific-asia conference on knowledge discovery and data mining*, p. 222–233. London, UK, UK, Springer-Verlag. Consulté sur http://dl.acm.org/citation.cfm?id=646419.693650

Kalousis A., Hilario M. (2001b). Model selection via meta-learning: a comparative study. *International Journal on Artificial Intelligence Tools*, vol. 10, n° 04, p. 525–554.

Kuhn H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, vol. 2, n° 1-2, p. 83–97.

Leite R., Brazdil P., Vanschoren J. (2012). Selecting classification algorithms with active testing. In *Machine learning and data mining in pattern recognition*, p. 117–131. Springer.

Leyva E., Gonzalez A., Perez R. (2015). A set of complexity measures designed for applying meta-learning to instance selection. *Knowledge and Data Engineering, IEEE Transactions on*, vol. 27, n° 2, p. 354–367.

Long F., Peng H., Ding C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, n° 8, p. 1226–1238.

Michie D., Spiegelhalter D. J., Taylor C. C. (1994). *Machine learning, neural and statistical classification*. Upper Saddle River, NJ, USA, Ellis Horwood.

Ntoutsi I., Kalousis A., Theodoridis Y. (2008). A general framework for estimating similarity of datasets and decision trees: exploring semantic similarity of decision trees. In *Siam international conference on data mining*, p. 810–821.

Peng Y., Flach P. A., Brazdil P., Soares C. (2002). Decision tree-based data characterization for meta-learning. *IDDM-2002*, p. 111.

Pfahringer B., Bensusan H., Giraud-Carrier C. (2000). Tell me who can learn you and i can tell you who you are: Landmarking various learning algorithms. In *Proceedings of the* 17^{th} *international conference on machine learning*, p. 743–750.

Raynaut W., Soule-Dupuy C., Valles-Parlangeau N. (2016, novembre). Une approche par dissimilarité pour la caractérisation de jeux de données. In 32^{e} *conférence sur la gestion de données - principes, technologies et applications (bda).*

Serban F. (2013). *Toward effective support for data mining using intelligent discovery assistance*. Thèse de doctorat non publiée.

Smid J. (2016). *Computational intelligence methods in metalearning*. Thèse de doctorat non publiée.

Sun Q., Pfahringer B. (2013). Pairwise meta-rules for better meta-learning-based algorithm ranking. *Machine learning*, vol. 93, n° 1, p. 141–161.

Sun Q., Pfahringer B., Mayo M. (2012). Full model selection in the space of data mining operators. In *Proceedings of the 14th annual conference companion on genetic and evolutionary computation*, p. 1503–1504.

Todorovski L., Brazdil P., Soares C. (2000). Report on the experiments with feature selection in meta-level learning. In *Proceedings of the pkdd-00 workshop on data mining, decision support, meta-learning and ilp: forum for practical problem presentation and prospective solutions*, p. 27–39.

Uno T., Asai T., Uchida Y., Arimura H. (2004). An efficient algorithm for enumerating closed patterns in transaction databases. In *Discovery science*, p. 16–31.

Vanschoren J., Blockeel H., Pfahringer B., Holmes G. (2012). Experiment databases. *Machine Learning*, vol. 87, n° 2, p. 127–158.

Vilalta R., Drissi Y. (2002, octobre). A perspective view and survey of meta-learning. *Artificial Intelligence Review*, vol. 18, n° 2, p. 77–95.

Wang L., Sugiyama M., Yang C., Hatano K., Feng J. (2009). Theory and algorithm for learning with dissimilarity functions. *Neural computation*, vol. 21, n° 5, p. 1459–1484.

Wistuba M., Schilling N., Schmidt-Thieme L. (2015). Learning data set similarities for hyperparameter optimization initializations. In *Metasel@ pkdd/ecml*, p. 15–26. Consulté sur http://ceur-ws.org/Vol-1455/#paper-04

Yoo A. B., Jette M. A., Grondona M. (2003). Slurm: Simple linux utility for resource management. In *Job scheduling strategies for parallel processing*, p. 44–60.

Zakova M., Kremen P., Zelezny F., Lavrac N. (2011). Automating knowledge discovery workflow composition through ontology-based planning. *Automation Science and Engineering, IEEE Transactions on*, vol. 8, n° 2, p. 253–264.