Apprentissage de bonnes similarités pour la classification linéaire parcimonieuse*

Aurélien Bellet, Amaury Habrard, Marc Sebban

Laboratoire Hubert Curien UMR CNRS 5516,
University of Jean Monnet, 42000 Saint-Etienne Cedex 2, France
{aurelien.bellet,amaury.habrard,marc.sebban}@univ-st-etienne.fr

Résumé : Le rôle crucial joué par les métriques au sein des processus d'apprentissage automatique a donné lieu ces dernières années à un intérêt croissant pour l'optimisation de fonctions de distances ou de similarités. La plupart des approches de l'état de l'art visent à apprendre une distance de Mahalanobis, devant satisfaire la contrainte de semi-définie positivité (SDP), exploitée in fine dans un algorithme *local* de type plus-proches-voisins. Cependant, aucun résultat théorique n'établit le lien entre les métriques apprises et leur comportement en classification. Dans cet article, nous exploitons le cadre formel des *bonnes similarités* pour proposer un algorithme d'apprentissage de similarité linéaire, optimisée dans un espace kernélisé. Nous montrons que la similarité apprise, ne requérant pas d'être SDP, possède des propriétés théoriques de stabilité permettant d'établir une borne en généralisation. Les expérimentations menées sur plusieurs jeux de données confirment son efficacité par rapport à l'état de l'art. **Mots-clés** : Apprentissage de similarité, apprentissage parcimonieux, classification linéaire.

1. Introduction

La notion de similarité (ou de distance) joue un rôle important dans beaucoup de problèmes d'apprentissage automatique, tels que la classification, le clustering ou le ranking. Cela explique l'abondance de travaux qui étudient, sous un angle pratique ou théorique, les propriétés qui font qu'une similarité est jugée "bonne". Étant donné qu'il est souvent difficile et fastidieux de construire manuellement une bonne fonction pour un problème donné, de nombreux travaux proposent des méthodes pour les apprendre automatiquement à partir de données étiquetées, donnant lieu à l'émergence de

*Ce travail est partiellement financé par le projet ANR LAMPADA 09-EMER-007-02.

l'*apprentissage supervisé de similarités et de métriques*. Parmi ces approches, l'apprentissage de distance de Mahalanobis a suscité un large intérêt (Schultz & Joachims, 2003; Shalev-Shwartz *et al.*, 2004; Davis *et al.*, 2007; Jain *et al.*, 2008; Weinberger & Saul, 2009; Ying *et al.*, 2009). Ces méthodes optimisent une matrice semi-définie positive (SDP) qui projette linéairement les données dans un nouvel espace, dans lequel la distance euclidienne est calculée. D'autres travaux proposent l'apprentissage de similarités cosinus (Qamar, 2010), ainsi que des distances locales (Frome *et al.*, 2007) et des similarités bilinéaires (Chechik *et al.*, 2009) dans le cadre de la recherche d'image.

En général, l'apprentissage supervisé de similarités et de métriques est basé sur l'intuition que la distance entre points de même classe doit être petite tandis que la distance entre points de classes différentes doit être grande. Appliquant cette idée, les approches de l'état de l'art cherchent la fonction qui satisfait le mieux ces *contraintes locales entre paires d'exemples*. La fonction apprise est ensuite utilisée dans le cadre des k-plus-proches-voisins (k-PPV), dont la règle de décision est basée sur un voisinage local. La performance obtenue est souvent meilleure que celle d'un k-PPV avec une fonction standard (par exemple la distance euclidienne). Cependant, il n'est pas évident que ces procédures d'apprentissage locales soient appropriées pour une utilisation dans des classifieurs globaux, tels que des séparateurs linéaires.

Récemment, Balcan et al. (2006; 2008) ont développé une théorie de l'apprentissage de classifieurs linéaires à partir de fonctions de similarité. Ils proposent une définition de (ϵ, γ, τ)-*bonne fonction de similarité*, sans contrainte SDP, qui requiert (en simplifiant) que pour la plupart des points, la similarité *moyenne* à *certains* points de même classe soit plus grande qu'à *certains* points de classe différente. Si l'on suppose que cette propriété est satisfaite, alors on dispose de garanties en généralisation sur l'erreur d'un classifieur linéaire parcimonieux dans l'espace induit par cette fonction de similarité.

Dans ce papier, nous utilisons la notion d'(ϵ, γ, τ)-goodness pour développer un algorithme d'apprentissage de similarités qui jouit de garanties en généralisation. Cette nouvelle approche a plusieurs avantages : (i) elle est adaptée aux classifieurs linéaires, (ii) justifiée théoriquement, (iii) elle ne requiert pas la semi-définie positivité de la similarité, et (iv) est dans un sens moins restrictive que l'apprentissage par paires. Nous formulons le problème d'apprentissage d'une bonne fonction de similarité comme un programme quadratique convexe qui optimise une similarité bilinéaire. De plus, en utilisant l'astuce KCPA (Chatpatanasiri *et al.*, 2010), nous sommes capables de "kerneliser" notre méthode et ainsi d'apprendre une similarité dans l'espace (potentielle-

ment non linéaire) induit par une fonction noyau. Nous montrons que notre approche a la propriété de stabilité uniforme (Bousquet & Elisseeff, 2002) et est donc consistante, et nous développons des garanties en généralisation qui portent sur l'(ϵ, γ, τ)-goodness de la similarité apprise. Enfin, nous proposons une étude expérimentale sur quatre jeux de données et comparons notre méthode à un algorithme d'apprentissage de métriques très répandu. Cette étude démontre les bonnes performances de notre approche et met en évidence le fait qu'elle est rapide, résistante au sur-apprentissage et induit des modèles très parcimonieux, ce qui la rend adaptée à des données de grande dimension.

La suite de ce papier est organisée comme suit. La Section 2. rappelle les principaux travaux en apprentissage de similarités et de métriques, ainsi que la théorie des (ϵ, γ, τ)-bonnes fonctions de similarité. La Section 3. présente notre approche ainsi que l'astuce KPCA utilisée pour la kerneliser. La Section 4. propose une analyse théorique de notre approche, menant à des garanties en généralisation. Enfin, la Section 5. comprend une étude expérimentale sur plusieurs jeux de données.

2. Notations et État de l'Art

Les vecteurs sont dénotés par des lettres minuscules en gras (par exemple, $\mathbf{x}$) et les matrices par des lettres majuscules en gras (par exemple, $\mathbf{A}$). On considère des points étiquetés $\mathbf{z} = (\mathbf{x}, \ell)$ issus d'une distribution inconnue P sur $\mathbb{R}^d \times \{-1, 1\}$. Une fonction de similarité est définie par $K : \mathbb{R}^d \times \mathbb{R}^d \to [-1, 1]$. On note la norme euclidienne $\|\cdot\|_2$ et la norme de Frobenius par $\|\cdot\|_{\mathcal{F}}$. Enfin, $[1 - c]_+ = \max(0, 1 - c)$ dénote la perte hinge.

2.1. Apprentissage de Métriques et de Similarités

Un grand nombre de travaux sur l'apprentissage de métriques et de similarités porte sur l'apprentissage des paramètres d'une distance de Mahalanobis. La distance de Mahalanobis (au carré), définie par $d^2_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = (\mathbf{x} - \mathbf{x}')^T\mathbf{M}(\mathbf{x}-\mathbf{x}')$, est paramétrisée par la matrice SDP $\mathbf{M} \in \mathbb{R}^{d\times d}$. Cette contrainte SDP assure que $d_{\mathbf{M}}$ est une (pseudo) métrique, ce qui permet des accélérations des k-PPV basées notamment sur l'inégalité triangulaire. Les différentes méthodes de la littérature diffèrent principalement par le choix de la fonction objective (ou de la fonction de perte) et du terme de régularisation. Par exemple, Schultz & Joachims (2003) forcent les exemples à être plus près des exemples de même classe que des exemples de classe différente par une certaine marge.

Weinberger & Saul (2009) définissent une fonction objective liée à l'erreur des k-PPV sur l'ensemble d'apprentissage. Davis et al. (2007) régularisent avec la divergence LogDet (qui impose automatiquement la contrainte SDP) tandis que Ying et al. (2009) utilisent la norme (2,1) qui favorisent l'apprentissage d'une matrice $\mathbf{M}$ de rang faible. Il existe aussi des méthodes d'apprentissage en ligne, comme POLA (Shalev-Shwartz *et al.*, 2004) et LEGO (Jain *et al.*, 2008). L'aspect le plus coûteux de beaucoup de ces approches est la satisfaction de la contrainte SDP, bien que certaines méthodes parviennent à réduire ce coût de calcul en développant des solveurs spécifiques.

Certains travaux portent sur l'apprentissage d'autres types de distances ou de similarités. Qamar (2010) optimise une similarité cosinus pour traiter des tâches de recherche d'information. Dans le domaine de la reconnaissance d'image, Frome et al. (2007) apprennent une distance locale pour chaque exemple d'apprentissage, tandis que Chechik et al. (2009) proposent une procédure d'apprentissage en ligne de similarité bilinéaire.

L'information utilisée en apprentissage supervisé de métriques et de similarités est de deux types : (i) des contraintes basées sur des paires d'exemples : $\mathbf{x}$ et $\mathbf{x}'$ doivent être similaires (ou dissimilaires), et (ii) des contraintes basées sur des triplets d'exemples : $\mathbf{x}$ doit être plus similaire à $\mathbf{x}'$ qu'à $\mathbf{x}''$. Notons que les deux types de contraintes peuvent être construits à partir de données étiquetées. L'objectif est alors de trouver la métrique ou la similarité qui satisfait le mieux ces contraintes.

Toutes les méthodes présentées ci-dessus sont en général utilisées dans le contexte des plus-proches-voisins (et parfois en clustering). Cela est dû au fait que les contraintes basées sur des paires ou des triplets sont faciles à obtenir et que les optimiser a du sens dans le cadre des k-PPV ou du clustering, qui sont des algorithmes basés sur des voisinages locaux. Dans le contexte de classifieurs globaux, comme les séparateurs linéaires, il n'est pas certain que ces contraintes locales soient appropriées. La théorie présentée dans la section suivante fait justement le lien entre les propriétés d'une fonction de similarité et ses performance en classification linéaire, ce qui ouvre la porte à l'apprentissage de similarités pour améliorer les séparateurs linéaires.

2.2. Apprentissage à partir de Bonnes Fonctions de Similarité

Dans des travaux récents, Balcan et al. (2006; 2008) ont introduit une nouvelle théorie de l'apprentissage à partir de *bonnes fonctions de similarité*, basée sur la définition suivante.

Définition 1 (Balcan et al., 2008)
Une fonction de similarité K est une (ϵ, γ, τ)-bonne fonction de similarité en perte hinge *pour un problème d'apprentissage P s'il existe une fonction indicatrice $R(\mathbf{x})$ caractérisant un ensemble de "points raisonnables" telle que les conditions suivantes sont satisfaites :*

1. $\mathbf{E}_{(\mathbf{x},\ell)\sim P}\left[[1 - \ell g(\mathbf{x})/\gamma]_+\right] \leq \epsilon$, *où* $g(\mathbf{x}) = \mathbf{E}_{(\mathbf{x}',\ell')\sim P}[\ell' K(\mathbf{x}, \mathbf{x}')|R(\mathbf{x}')]$,
2. $\Pr_{\mathbf{x}'}[R(\mathbf{x}')] \geq \tau$.

Si l'on réfléchit à cette définition en terme de nombre de violations de marge, on peut interpréter la première condition comme *une proportion ϵ d'exemples $\mathbf{x}$ sont en moyenne 2γ plus similaires aux points raisonnables de même classe qu'aux points raisonnables de classe différente* et la seconde condition comme *une proportion au moins τ des exemples doivent être raisonnables*.

La Définition 1 est intéressante à trois égards. Premièrement, elle n'impose aucune contrainte SDP (ou même de symétrie) sur la fonction de similarité. Deuxièmement, les contraintes doivent être satisfaites seulement en moyenne, au contraire des contraintes classiques basées sur des paires ou des triplets. Troisièmement, satisfaire la Définition 1 est suffisant pour bien apprendre, comme le montre le Théorème 1.

Théorème 1 (Balcan et al., 2008)
Soit K une (ϵ, γ, τ)-bonne fonction de similarité en perte hinge pour un problème d'apprentissage P. Pour tout $\epsilon_1 > 0$ et $0 \leq \delta \leq \gamma\epsilon_1/4$, soit $S = \{\mathbf{x}'_1, \ldots, \mathbf{x}'_{\mathbf{d_{land}}}\}$ un ensemble (potentiellement non étiqueté) de points landmarks tirés selon P. Considérons l'application $\phi^S : \mathbb{R}^d \to \mathbb{R}^{d_{land}}$ définie par : $\phi^S_i(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}'_\mathbf{i})$, $i \in \{1, \ldots, d_{land}\}$. Alors, avec une probabilité au moins $1 - \delta$ sur l'ensemble aléatoire S, la distribution induite $\phi^S(P)$ dans $\mathbb{R}^{d_{land}}$ admet un séparateur linéaire $\boldsymbol{\alpha}$ d'erreur au plus $\epsilon + \epsilon_1$ à la marge γ.

Ainsi, si l'on dispose d'une (ϵ, γ, τ)-bonne fonction de similarité pour un problème P et d'assez de points, alors avec une grande probabilité il existe un séparateur linéaire $\boldsymbol{\alpha}$ de faible erreur dans l'espace explicite induit par ϕ, qui est essentiellement l'espace des similarités aux d_{land} points landmarks. Comme Balcan et al. le soulignent, en utilisant d_u points landmarks (potentiellement non étiquetés) et d_l points étiquetés, on peut apprendre efficacement ce sépa-

rateur $\boldsymbol{\alpha} \in \mathbb{R}^{d_u}$ en résolvant le programme linéaire suivant :[1]

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{d_l} \left[1 - \sum_{j=1}^{d_u} \alpha_j \ell_i K(\mathbf{x_i}, \mathbf{x'_j}) \right]_+ + \lambda \|\boldsymbol{\alpha}\|_1. \quad (1)$$

L'Equation (1) est proche d'un SVM avec une régularisation L_1 (Zhu *et al.*, 2003) et un "empirical similarity map" (Balcan & Blum, 2006), et peut-être résolue efficacement. La régularisation L_1 induit de la parcimonie (coordonnées à zéro) dans $\boldsymbol{\alpha}$, ce qui permet d'accélérer la classification. Il est possible de contrôler le degré de parcimonie en jouant sur le paramètre λ (plus λ est grand, plus $\boldsymbol{\alpha}$ est parcimonieux).

Pour résumer, l'erreur du séparateur linéaire dépend théoriquement de la capacité de la similarité à satisfaire la Définition 1. Cependant, il est probable que, pour certains problèmes, les similarités standards ne satisfassent pas la Définition 1 à un degré suffisant. Ainsi, Kar & Jain (2011) proposent d'adapter automatiquement le critère de goodness au problème traité. Dans ce papier, nous suivons une approche différente : nous utilisons la Définition 1 comme une nouvelle fonction objective, justifiée théoriquement, pour l'apprentissage de similarités.

3. Apprentissage de Bonnes Similarités pour la Classification Linéaire

Dans ce papier, on s'intéresse à la *similarité bilinéaire* $K_{\mathbf{A}} = \mathbf{x}^T \mathbf{A} \mathbf{x}'$, paramétrée par la matrice $\mathbf{A} \in \mathbb{R}^{d \times d}$, qui n'est pas nécessairement SDP ni symétrique. Cette forme de similarité a donné lieu à de bons résultats dans le cadre de l'apprentissage en ligne de similarités pour la recherche d'image (Chechik *et al.*, 2009). Il est intéressant de noter que $K_{\mathbf{A}}$ a l'avantage d'être calculable efficacement quand les exemples $\mathbf{x}$ et $\mathbf{x}'$ sont des vecteurs parcimonieux. Afin de satisfaire la contrainte $K_{\mathbf{A}} \in [-1, 1]$, on suppose que les exemples sont normalisés tels que $||\mathbf{x}||_2 \leq 1$, et l'on exige que $||\mathbf{A}||_{\mathcal{F}} \leq 1$.

3.1. Formulation du Problème d'Apprentissage de Similarités

On souhaite apprendre la matrice $\mathbf{A}$ qui optimise l'(ϵ, γ, τ)-goodness de $K_{\mathbf{A}}$. Pour cela, on dispose d'un ensemble d'apprentissage de N_T points éti-

[1] La formulation originale proposée par Balcan et al. (2008) comprend une contrainte L_1 au lieu d'une régularisation L_1, mais les deux formulations sont équivalentes (il existe toujours une valeur du paramètre de la première telle que l'ensemble des solutions est le même que celui de la deuxième, et vice versa).

quetés $T = \{\mathbf{z_i} = (\mathbf{x_i}, \ell_i)\}_{i=1}^{N_T}$ ainsi que d'un ensemble de N_R points raisonnables étiquetés $R = \{\mathbf{z_k} = (\mathbf{x_k}, \ell_k)\}_{k=1}^{N_R}$. En pratique, R est un sous-ensemble de T, avec $N_R = \hat{\tau} N_T$ ($\hat{\tau} \in]0, 1]$).

Nous définissons maintenant formellement notre approche d'apprentissage d'(ϵ, γ, τ)-bonne similarité K_A :

$$\min_{\mathbf{A} \in \mathbb{R}^{d \times d}} \quad \frac{1}{N_T} \sum_{i=1}^{N_T} V(\mathbf{A}, \mathbf{z_i}, R) \quad + \quad \beta \|\mathbf{A}\|_{\mathcal{F}}^2 \tag{2}$$

où $V(\mathbf{A}, \mathbf{z_i}, R) = [1 - \ell_i \frac{1}{\gamma N_R} \sum_{k=1}^{N_R} \ell_k K_{\mathbf{A}}(\mathbf{x_i}, \mathbf{x_k})]_+$, γ est la marge et β le paramètre de régularisation. Il est important de voir que cette formulation découle directement de la Définition 1 : étant donnés une marge γ et un ensemble de points raisonnables R, on minimise les violations de marge ϵ des exemples de T par rapport à R. Ainsi, $\hat{\epsilon} = \frac{1}{N_T} \sum_{i=1}^{N_T} V(\mathbf{A}, \mathbf{z_i}, R)$ est l'estimation empirique d'ϵ sur T, et $\hat{\tau}$ une estimation de τ.

Cette formulation est radicalement différente de celles basées sur des paires ou des triplets, classiquement utilisées par les approches d'apprentissage de métriques et de similarités. Dans un sens, elle est plus facile à satisfaire puisque les contraintes sont basées sur une *moyenne* de scores de similarité par rapport à un ensemble de points R, au lieu d'une seule paire ou triplet. De plus, comme R est le même pour chaque exemple d'apprentissage, la similarité apprise est plutôt globale que locale, ce qui peut être un avantage dans certaines situations. En outre, notre formulation possède plusieurs propriétés intéressantes : (i) Il s'agit d'un programme quadratique convexe, qui peut être résolu efficacement par des solveurs standards. La programmation semi-définie, très coûteuse, n'est pas nécessaire, à l'inverse d'un grand nombre de méthodes d'apprentissage de distance de Mahalanobis. (ii) On ne définit qu'une seule contrainte par exemple d'apprentissage (au lieu d'une contrainte par paire ou triplet), c'est-à-dire un nombre total de N_T contraintes. En utilisant une variable ressort par contrainte, le problème a seulement $N_T + d^2$ variables. (iii) La taille de R n'affecte pas la complexité des contraintes. (iv) Si $\mathbf{x_i}$ est parcimonieux, la contrainte correspondante est parcimonieuse également (certaines variables ont un coefficient de 0).

3.2. Kernelisation de l'Apprentissage de Similarités

La formulation introduite dans la section précédente est théoriquement fondée (par rapport à la théorie de Balcan et al.) et présente, comme nous

le verrons dans la section suivante, des garanties en généralisation. De plus, elle a l'avantage d'être très simple : on apprend une similarité linéaire globale et on l'utilise pour construire un classifieur linéaire global. Afin d'obtenir des similarités (et ainsi des classifieurs) plus puissants, nous suggérons de kerneliser l'approche en apprenant dans l'espace potentiellement non linéaire induit par une fonction noyau. La kernelisation permet aux classifieurs linéaires tels que les SVM ou encore à certains algorithmes d'apprentissage de distance de Mahalanobis (par exemple, Shalev-Shwartz *et al.*, 2004; Davis *et al.*, 2007) d'apprendre des surfaces de décision ou des transformations non linéaires. Cependant, kerneliser un algorithme d'apprentissage de métriques n'est pas trivial : une nouvelle formulation du problème, où l'interaction avec les données est limitée à des produits scalaires, doit être trouvée. De plus, quand la kernelisation est possible, on apprend une matrice $N_T \times N_T$. Quand N_T est grand, la résolution du problème devient trop coûteuse, à moins d'appliquer une méthode de réduction de dimensionnalité.

Pour ces raisons, nous utilisons plutôt l'astuce KPCA, proposée récemment par Chatpatanasiri et al. (2010), qui donne un moyen simple de kerneliser un algorithme d'apprentissage de métriques et de faire de la réduction de dimensionnalité sans coût supplémentaire. L'idée est d'utiliser Kernel Principal Component Analysis (Schölkopf *et al.*, 1998) pour projeter les données dans un nouvel espace en utilisant une fonction noyau non linéaire, et de conserver uniquement un nombre limité de dimensions (celles qui capturent le mieux la variance globale des données). Les données sont ensuite projetées dans ce nouvel espace, et l'algorithme d'apprentissage de métriques peut être utilisé pour apprendre une métrique dans cet espace, sans modification. Chatpatanasiri et al. (2010) ont montré que l'astuce KPCA est théoriquement justifiée pour l'apprentissage de métriques et de similarités formulé sans contraintes, dont notre approche fait partie. Dans la suite de ce papier, on considérera uniquement la version kernelisée de notre algorithme.

De manière générale, la kernelisation d'un algorithme d'apprentissage de métriques peut causer ou accroître le sur-apprentissage, notamment quand les données sont peu nombreuses. Néanmoins, comme notre approche (la similarité et le classifieur) est linéaire et globale, on s'attend à ce qu'elle soit assez robuste à cet effet indésirable. La suite du papier va le confirmer doublement : expérimentalement dans la Section 5., mais aussi théoriquement avec la dérivation, dans la section suivante, de garanties en généralisation indépendantes de la dimension de l'espace de projection.

4. Analyse Théorique

Dans cette section, nous présentons une analyse théorique de notre approche. Le fruit de cette analyse est la borne en généralisation (Théorème 3) qui garantit la consistance de notre méthode et donc l'(ϵ, γ, τ)-goodness de la similarité apprise pour le problème considéré.

Dans notre approche, la similarité est optimisée par rapport à un ensemble R de points raisonnables qui est un sous-ensemble de l'ensemble d'apprentissage. Ainsi, ces points raisonnables peuvent ne pas suivre la même distribution que celle qui a servi à générer l'ensemble d'apprentissage. C'est pourquoi nous allons utiliser le cadre de la *stabilité uniforme* (Bousquet & Elisseeff, 2002) pour obtenir notre borne en généralisation.

4.1. Stabilité Uniforme

Schématiquement, un algorithme est *stable* si sa sortie ne change pas significativement quand on modifie légèrement l'ensemble d'apprentissage. Plus précisément, le supremum de cette variation doit être bornée par un terme en $\mathcal{O}(1/N_T)$.

Définition 2 (Bousquet & Elisseeff, 2002)
Un algorithme d'apprentissage a une stabilité uniforme en $\frac{\kappa}{N_T}$ par rapport à une fonction de perte $\mathcal{L}$, κ étant une constante positive, si

$$\forall T, \forall i, 1 \leq i \leq N_T, \sup_{\mathbf{z}} |\mathcal{L}(\mathbf{M}_T, \mathbf{z}) - \mathcal{L}(\mathbf{M}_{T^i}, \mathbf{z})| \leq \frac{\kappa}{N_T},$$

où $\mathbf{M}_T$ est le modèle appris à partir de l'ensemble T, $\mathbf{M}_{T^i}$ le modèle appris à partir de l'ensemble T^i. T^i est obtenu à partir de T en remplaçant le i^{eme} exemple $\mathbf{z}_i \in T$ par un autre exemple $\mathbf{z}'_i$ indépendant de T et tiré selon P. $\mathcal{L}(\mathbf{M}, \mathbf{z})$ est la perte pour un exemple $\mathbf{z}$.

Quand cette définition est satisfaite, Bousquet & Elisseeff (2002) ont prouvé la borne suivante, qui porte sur l'erreur en généralisation.

Théorème 2 (Bousquet & Elisseeff, 2002)
Soit $\delta > 0$ et $N_T > 1$. Pour tout algorithme ayant une stabilité uniforme κ/N_T et utilisant une fonction de perte bornée par 1, on a avec une probabilité au moins $1-\delta$:

$$L(\mathbf{M}_T) < \hat{L}_T(\mathbf{M}_T) + \frac{\kappa}{N_T} + (2\kappa + 1)\sqrt{\frac{\ln 1/\delta}{2N_T}},$$

$L(\mathbf{M}_T)$ est l'erreur en généralisation et $\hat{L}_T(\mathbf{M}_T)$ son estimation empirique sur T.

4.2. Borne en Généralisation

Pour alléger la notation, étant donnée une similarité bilinéaire $K_{\mathbf{A}}$, on dénote par $\mathbf{A}_R$ à la fois la similarité définie par la matrice $\mathbf{A}$ et son ensemble de points raisonnables associé (quand le contexte est clair, on omettra l'indice R). Étant donnée une similarité $\mathbf{A}_R$, V définit la perte pour un exemple, et l'on note l'erreur sur l'ensemble de la distribution par

$$L(\mathbf{A}_R) = \mathbb{E}_{\mathbf{z}=(\mathbf{x},l)\sim P} V(\mathbf{A}, \mathbf{z}, R). \tag{3}$$

L'erreur empirique sur l'ensemble T est définie par :

$$\hat{L}_T(\mathbf{A}_R) = \frac{1}{N_T} \sum_{i=1}^{N_T} V(\mathbf{A}, \mathbf{z}_i, R). \tag{4}$$

Adaptée à notre contexte, la propriété de stabilité uniforme que l'on doit prouver est la suivante :

$$\forall T, \forall i, \sup_{\mathbf{z}} |V(\mathbf{A}, \mathbf{z}, R) - V(\mathbf{A}^i, \mathbf{z}, R^i)| \leq \frac{\kappa}{N_T},$$

où $\mathbf{A}$ est apprise sur T et $R \subseteq T$, $\mathbf{A}^i$ est la matrice apprise sur T^i et $R^i \subseteq T^i$ est l'ensemble des points raisonnables associé à T^i. Notons que R et R^i sont de même taille et diffèrent sur au plus un exemple, en fonction de l'appartenance ou non de $\mathbf{z}_i$ ou $\mathbf{z}'_i$ à leur ensemble de points raisonnables respectifs. Pour simplifier la discussion, on considère V comme majorée par 1 (ce qui peut être obtenu facilement en divisant V par la constante $1 + \frac{1}{\gamma}$).

Pour montrer la stabilité uniforme de notre algorithme, nous avons besoin des résultats suivants.

Lemme 1
Pour tous exemples étiquetés $\mathbf{z} = (\mathbf{x}, \ell)$, $\mathbf{z}' = (\mathbf{x}', \ell')$ et tous modèles $\mathbf{A}_R$, $\mathbf{A}'_{R'}$, on a les propriétés suivantes :

P1 : $|K_{\mathbf{A}}(\mathbf{x}, \mathbf{x}')| \leq 1$,

P2 : $|K_{\mathbf{A}}(\mathbf{x}, \mathbf{x}') - K_{\mathbf{A}'}(\mathbf{x}, \mathbf{x}')| \leq \|\mathbf{A} - \mathbf{A}'\|_{\mathcal{F}}$,

P3 : $|V(\mathbf{A}, \mathbf{z}, R) - V(\mathbf{A}', \mathbf{z}, R')| \leq 1 |\frac{\sum_{k=1}^{N_R} \ell_k K_{\mathbf{A}}(\mathbf{x}, \mathbf{x}_k)}{\gamma N_R} - \frac{\sum_{j=1}^{N_{R'}} \ell'_k K_{\mathbf{A}'}(\mathbf{x}, \mathbf{x}'_k)}{\gamma N_{R'}}|$
(propriété de 1-admissibilité de V).

Preuve $P1$ découle directement de $|K_{\mathbf{A}}(\mathbf{x},\mathbf{x}')| \leq \|\mathbf{x}\|_2\|\mathbf{A}\|_{\mathcal{F}}\|\mathbf{x}'\|_2$, de la normalisation des exemples $\|\mathbf{x}\|_2 \leq 1$ et de la condition sur les matrices $\|\mathbf{A}\|_{\mathcal{F}} \leq 1$.

Pour $P2$, on note que $|K_{\mathbf{A}}(\mathbf{x},\mathbf{x}') - K_{\mathbf{A}'}(\mathbf{x},\mathbf{x}')| = |K_{\mathbf{A}-\mathbf{A}'}(\mathbf{x},\mathbf{x}')|$, puis on utilise la normalisation $\|\mathbf{x}\|_2 \leq 1$.

$P3$ découle directement de $|\ell| = 1$ et de la propriété de 1-lipschitzité de la perte hinge : $|[X]_+ - [Y]_+| \leq |X - Y|$. □

Soit $F_T = \frac{1}{N_T}\sum_{i=1}^{N_T} V(\mathbf{A},\mathbf{z_i},R) + \beta\|\mathbf{A}\|_{\mathcal{F}}^2$ pour un ensemble d'apprentissage T et un ensemble de points raisonnables $R \subseteq T$ donnés. Nous avons besoin du lemme suivant, qui borne la déviation entre les matrices $\mathbf{A}$ et $\mathbf{A}^i$.

Lemme 2
Pour tous modèles $\mathbf{A}$ *et* $\mathbf{A^i}$*, respectivement minimiseurs de* F_T *et* F_{T^i}*, on a :*

$$\|\mathbf{A} - \mathbf{A^i}\|_{\mathcal{F}} \leq \frac{1}{\beta N_T \gamma}.$$

Preuve On suit la construction du Lemme 20 de (Bousquet & Elisseeff, 2002), en omettant quelques détails pour des raisons de place. Soit $\Delta\mathbf{A} = \mathbf{A^i} - \mathbf{A}$ et $0 \leq t \leq 1$. On note

$$M_1 = \|\mathbf{A}\|_{\mathcal{F}}^2 - \|\mathbf{A} + t\Delta\mathbf{A}\|_{\mathcal{F}}^2 + \|\mathbf{A}^i\|_{\mathcal{F}}^2 - \|\mathbf{A}^i - t\Delta\mathbf{A}\|_{\mathcal{F}}^2$$

et $M_2 = \frac{1}{\beta N_T}(\hat{L}_T(\mathbf{A}_R) - \hat{L}_T((\mathbf{A}+t\Delta\mathbf{A})_R) + \hat{L}_{T^i}((\mathbf{A}+t\Delta\mathbf{A})_R) - \hat{L}_{T^i}(\mathbf{A}_R))$. En utilisant le fait que F_T et F_{T^i} sont des fonctions convexes, que $\mathbf{A}$ et $\mathbf{A}^i$ sont leur minimiseurs respectifs et la propriété $P3$, on obtient :

$$M_1 \leq M_2.$$

En fixant $t = 1/2$, on a $M_1 = \|\mathbf{A} - \mathbf{A}^i\|_{\mathcal{F}}^2$, et en utilisant $P3$ et la normalisation $\|\mathbf{x}\|_2 \leq 1$, on obtient :

$$M_2 \leq \frac{1}{\beta N_T \gamma}(\|\frac{1}{2}\Delta\mathbf{A}\|_{\mathcal{F}} + \|-\frac{1}{2}\Delta\mathbf{A}\|_{\mathcal{F}}) = \frac{\|\mathbf{A} - \mathbf{A}^i\|_{\mathcal{F}}}{\beta N_T \gamma}.$$

Cela nous mène à l'inégalité $\|\mathbf{A} - \mathbf{A}^i\|_{\mathcal{F}}^2 \leq \frac{\|\mathbf{A}-\mathbf{A}^i\|_{\mathcal{F}}}{\beta N_T \gamma}$ à partir de laquelle le Lemme 2 est obtenu directement. □

Nous avons maintenant tous les résultats nécessaires pour prouver la stabilité uniforme de notre algorithme.

Lemme 3
Soit N_T et N_R respectivement le nombre d'exemples d'apprentissage et le nombre de points raisonnables, tels que $N_R = \hat{\tau} N_T$ avec $\hat{\tau} \in]0,1]$. Notre algorithme a une stabilité uniforme en $\frac{\kappa}{N_T}$ avec $\kappa = \frac{1}{\gamma}(\frac{1}{\beta\gamma} + \frac{2}{\hat{\tau}}) = \frac{\hat{\tau}+2\beta\gamma}{\hat{\tau}\beta\gamma^2}$, où β est le paramètre de régularisation et γ la marge du Problème 2.

Preuve Pour tout ensemble T de taille N_T, pour tout $1 \leq i \leq N_T$ et tous exemples étiquetés $\mathbf{z} = (\mathbf{x}, \ell), \mathbf{z}'_i = (\mathbf{x}_i, \ell'_i) \sim P$:

$$
\begin{aligned}
&|V(\mathbf{A}, \mathbf{z}, R) - V(\mathbf{A}^i, \mathbf{z}, R^i)| \\
&\leq \left| \frac{1}{\gamma N_R} \sum_{k=1}^{N_R} \ell_k K_{\mathbf{A}}(\mathbf{x}, \mathbf{x}_k) - \frac{1}{\gamma N_{R^i}} \sum_{k=1}^{N_{R^i}} \ell_k K_{\mathbf{A}^i}(\mathbf{x}, \mathbf{x}_k) \right| \\
&= \left| \frac{1}{\gamma N_R} \left(\left(\sum_{k=1, k\neq i}^{N_R} (\ell_k K_{\mathbf{A}}(\mathbf{x}, \mathbf{x}_k) - K_{\mathbf{A}^i}(\mathbf{x}, \mathbf{x}_k)) \right) + \right. \right. \\
&\qquad \left. \left. \ell_i K_{\mathbf{A}}(\mathbf{x}, \mathbf{x}_i) - \ell'_i K_{\mathbf{A}^i}(\mathbf{x}, \mathbf{x}'_i) \right) \right| \\
&\leq \frac{1}{\gamma N_R} \left(\left(\sum_{k=1, k\neq i}^{N_R} (|\ell_k| \|\mathbf{A} - \mathbf{A}^i\|_{\mathcal{F}}) \right) + \right. \\
&\qquad \left. |\ell_i K_{\mathbf{A}^i}(\mathbf{x}, \mathbf{x}_i)| + |\ell'_i K_{\mathbf{A}}(\mathbf{x}, \mathbf{x'}_i)| \right) \\
&\leq \frac{1}{\gamma N_R} (\frac{N_R - 1}{\beta N_T \gamma} + 2) \leq \frac{1}{\gamma N_R} (\frac{N_R}{\beta N_T \gamma} + 2).
\end{aligned}
$$

La première inégalité découle de $P3$. La seconde découle du fait que R et R^i diffèrent sur au plus un élément, qui correspond à l'exemple $\mathbf{z}_i$ dans R et l'exemple $\mathbf{z}'_i$ remplaçant $\mathbf{z}_i$ dans R^i. Les dernières inégalités sont obtenues en utilisant l'inégalité triangulaire, $P1$, $P2$, le Lemme 2, et le fait que les étiquettes appartiennent à $\{-1, 1\}$. Enfin, comme $N_R = \hat{\tau} N_T$, on obtient

$$|V(\mathbf{A}, \mathbf{z}, R) - V(\mathbf{A}^i, \mathbf{z}, R^i)| \leq \frac{1}{\gamma N_T} (\frac{1}{\beta\gamma} + \frac{2}{\hat{\tau}}).$$

□

En combinant le Théorème 2 et le Lemme 2, on obtient notre borne en généralisation.

Théorème 3
Soit $\gamma > 0$, $\delta > 0$ et $N_T > 1$. Avec une probabilité au moins $1 - \delta$, pour tout modèle $\mathbf{A}_R$ appris avec le Problème 2, on a :

$$L \leq \hat{L}_T + \frac{1}{N_T}\left(\frac{\hat{\tau} + 2\beta\gamma}{\hat{\tau}\beta\gamma^2}\right) + \left(\frac{2(\hat{\tau} + 2\beta\gamma)}{\hat{\tau}\beta\gamma^2} + 1\right)\sqrt{\frac{\ln 1/\delta}{2N_T}}.$$

Ce théorème met en lumière d'importantes propriétés de notre méthode. Tout d'abord, elle converge en $\mathcal{O}(1/\sqrt{N_T})$, ce qui est standard dans le cadre de la stabilité uniforme. De plus, cette convergence est indépendante de la dimension de la matrice $\mathbf{A}$ apprise par la méthode, et donc de la dimension des données. C'est la conséquence du fait que la norme de Frobenius de $\mathbf{A}$ est bornée par une constante. Un autre point très important est que le Théorème 3 borne en fait l'(ϵ, γ, τ)-goodness en généralisation de la similarité apprise, et ainsi l'erreur du classifieur linéaire construit à partir de cette similarité. En effet, L correspond au terme d'erreur ϵ de Balcan et al. (Définition 1).

5. Étude Expérimentale

Nous proposons une étude comparative de notre méthode et d'une méthode réputée d'apprentissage de distance de Mahalanobis : Large Margin Nearest Neighbor[2] (LMNN) de Weinberger & Saul (2009). Nous menons cette étude expérimentale sur quatre jeux de données de différents domaines et de difficulté variable, dont la plupart sont issus de l'UCI Machine Learning Repository.[3] Leurs caractéristiques sont résumées dans le Tableau 1. Breast, Ionosphere et Pima ont été utilisés à maintes reprises pour évaluer les approches d'apprentissage de métriques.

5.1. Protocole Expérimental

On compare les méthodes suivantes : (i) K_I, la similarité bilinéaire avec $A = I$ (proche de la similarité cosinus) comme base, (ii) K_A, notre similarité apprise, (iii) LMNN dans l'espace original, et (iv) LMNN dans l'espace KPCA.[4]

[2] Code Matlab disponible sur : `http://www.cse.wustl.edu/~kilian/code/code.html`

[3] Voir `http://archive.ics.uci.edu/ml/`

[4] K_I et LMNN sont normalisés pour s'assurer que leurs valeurs appartiennent à $[-1, 1]$.

TAB. 1: Caractéristiques des quatre jeux de données utilisés.

	BREAST	IONO.	RINGS	PIMA
# training examples	488	245	700	537
# test examples	211	106	300	231
# dimensions	9	34	2	8
# dim. after KPCA	27	102	8	24

Pour générer un nouvel espace de projection avec KPCA, on utilise le noyau Gaussien en fixant le paramètre σ à la moyenne des distances euclidiennes entre les exemples d'apprentissage (une heuristique classique, utilisée par exemple par Kar & Jain (2011)). Idéalement, on aimerait projeter les données dans un espace de dimension maximale (c'est-à-dire égale au nombre d'exemples d'apprentissage), mais pour garder des temps d'exécution acceptables on ne garde que trois fois la dimension originale (quatre fois pour Rings car ce jeu de données est seulement de dimension 2).[5]

Tous les attributs sont normalisés pour que leurs valeurs soient dans $[-1/d; 1/d]$ afin de garantir $\|\mathbf{x}\|_2 \leq 1$. On génère aléatoirement des partitions 70/30 des données, et on calcule la moyenne sur 100 exécutions. Les ensembles d'apprentissages sont partitionnés une nouvelle fois 70/30 pour le besoin de la validation. Les paramètres suivants sont tunés par cross-validation : $\beta, \gamma \in \{10^{-7}, \ldots, 10^{-2}\}$ et $\hat{\tau} \in \{0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 1\}$ pour notre méthode, et $\lambda \in \{10^{-3}, \ldots, 10^{2}\}$ pour chaque classifieur linéaire, en choisissant toujours la valeur qui offre le meilleur taux de classification. On fixe $k = 3$ et $\mu = 0.5$ pour LMNN, comme suggéré dans (Weinberger & Saul, 2009).

5.2. Résultats

Comme LMNN est optimisé pour un usage dans un k-PPV, on reporte les résultats obtenus avec le classifieur linéaire parcimonieux (1) suggéré par Balcan et al. (2008) (Tableau 2) mais aussi ceux obtenus en 3-PPV (Tableau 3). Le style gras met en évidence le meilleur taux de classification pour chaque jeu de données. En classification linéaire, notre méthode obtient des taux de classification supérieurs ou égaux aux autres similarités avec des modèles de

[5] En appliquant cette stratégie, la proportion de la variance capturée est supérieure à 90% pour tous les jeux de données.

TAB. 2: Taux de classification moyen (en police normale) et taille (en italique) des classifieurs linéaires construits à partir des similarités étudiées.

	BREAST	IONO.	RINGS	PIMA
K_I	96.57	89.81	**100.00**	75.62
	20.39	*52.93*	***18.20***	*25.93*
K_A	**96.93**	**93.05**	**100.00**	**76.07**
	1.00	***1.00***	***1.00***	***1.00***
LMNN	96.81	90.21	**100.00**	75.15
	9.98	*13.30*	***18.04***	*69.71*
LMNN KPCA	96.01	86.12	**100.00**	74.92
	8.46	*9.96*	***8.73***	*22.20*

TAB. 3: Taux de classification moyen des 3-PPV avec les similarités étudiées.

	BREAST	IONO.	RINGS	PIMA
K_I	96.71	83.57	**100.00**	72.78
K_A	**96.93**	**93.05**	**100.00**	**76.07**
LMNN	96.46	88.68	**100.00**	72.84
LMNN KPCA	96.23	87.13	**100.00**	73.50

taille 1, ce qui rend la classification très rapide car elle ne dépend que d'un score de similarité à un seul exemple d'apprentissage. Chose intéressante, K_A obtient aussi les meilleurs résultats en 3-PPV. De manière générale, LMNN a une tendance à sur-apprendre (surtout dans l'espace KPCA) car les contraintes basées sur des paires deviennent faciles à satisfaire quand la dimension augmente. L'extrême parcimonie des séparateurs linéaires et la résistance au sur-apprentissage de notre méthode viennent du fait que les contraintes optimisées portent sur une moyenne de scores de similarité par rapport à un ensemble de points commun à tous les exemples d'apprentissage.

6. Conclusion

Dans ce papier, nous avons présenté une approche nouvelle d'apprentissage de similarités qui vise à améliorer les classifieurs linéaires. Notre méthode tire partie à la fois de la théorie des (ϵ, γ, τ)-bonnes similarités de Balcan et al. et de l'astuce KPCA. Nous avons prouvé une borne en généralisation pour notre approche, basée sur la stabilité uniforme, qui est indépendante de la

dimension des données et donc du nombre de coordonnées sélectionnées par KPCA. Cette borne assure la “goodness” en généralisation de la similarité apprise, garantissant ainsi que celle-ci donne lieu à des classifieurs performants pour le problème considéré. L’étude expérimentale confirme cette propriété, et montre également que les similarités apprises génèrent des classifieurs bien plus parcimonieux que ceux obtenus avec d’autres similarités standards ou apprises. Cette caractéristique, combinée à l’indépendance par rapport à la dimension des données, rend notre approche très efficace. Les perspectives possibles sont : la kernelisation complète de l’algorithme, l’étude de l’influence d’autres termes de régularisation, ou encore la mise au point d’un algorithme d’apprentissage en ligne.

Références

BALCAN M.-F. & BLUM A. (2006). On a Theory of Learning with Similarity Functions. In *ICML*, p. 73–80.

BALCAN M.-F., BLUM A. & SREBRO N. (2008). Improved Guarantees for Learning via Similarity Functions. In *COLT*, p. 287–298.

BOUSQUET O. & ELISSEEFF A. (2002). Stability and Generalization. *JMLR*, **2**, 499–526.

CHATPATANASIRI R., KORSRILABUTR T., TANGCHANACHAIANAN P. & KIJSIRIKUL B. (2010). A new kernelization framework for Mahalanobis distance learning algorithms. *Neurocomputing*, **73**, 1570–1579.

CHECHIK G., SHALIT U., SHARMA V. & BENGIO S. (2009). An Online Algorithm for Large Scale Image Similarity Learning. In *NIPS*, p. 306–314.

DAVIS J. V., KULIS B., JAIN P., SRA S. & DHILLON I. S. (2007). Information-theoretic metric learning. In *ICML*, p. 209–216.

FROME A., SINGER Y., SHA F. & MALIK J. (2007). Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification. In *ICCV*, p. 1–8.

JAIN P., KULIS B., DHILLON I. S. & GRAUMAN K. (2008). Online Metric Learning and Fast Similarity Search. In *NIPS*, p. 761–768.

KAR P. & JAIN P. (2011). Similarity-based Learning via Data Driven Embeddings. In *NIPS*.

QAMAR A. (2010). *Generalized Cosine and Similarity Metrics : A supervised learning approach based on nearest-neighbors*. PhD thesis, University of Grenoble.

SCHÖLKOPF B., SMOLA A. & MÃ$\frac{1}{4}$LLER K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, **10**(1), 1299–1319.

SCHULTZ M. & JOACHIMS T. (2003). Learning a Distance Metric from Relative Comparisons. In *NIPS*.

SHALEV-SHWARTZ S., SINGER Y. & NG A. Y. (2004). Online and batch learning of pseudo-metrics. In *ICML*.

WEINBERGER K. Q. & SAUL L. K. (2009). Distance Metric Learning for Large Margin Nearest Neighbor Classification. *JMLR*, **10**, 207–244.

YING Y., HUANG K. & CAMPBELL C. (2009). Sparse Metric Learning via Smooth Optimization. In *NIPS*, p. 2214–2222.

ZHU J., ROSSET S., HASTIE T. & TIBSHIRANI R. (2003). 1-norm Support Vector Machines. In *NIPS*, volume 16, p. 49–56.