Une approche de génération de colonnes pour les machines à vecteurs de support

Emilio Carrizosa

Université de Séville (Espagne). ecarrizosa@us.es

Belen Mart 'ın-Barragan'

Université de Séville (Espagne). belmart@us.es

Dolorès Romero Morales

Université d'Oxford (Royaume-Uni). dolores.romero-morales@sbs.ox.ac.uk

23 mars 2006

Abstrait

La méthode SVM (Support Vector Machine), largement utilisée, s'est avérée donner de bons résultats dans Problèmes de classification supervisée. D'autres méthodes telles que les arbres de classification sont devenues plus plus populaires parmi les praticiens que les SVM grâce à leur interprétabilité, ce qui est un enjeu important en exploration de données.

Dans ce travail, nous proposons une méthode basée sur SVM qui détecte automatiquement les problèmes les plus importants. les variables prédictives et les valeurs qui sont critiques pour la classification. Sa capacité de classification est comparable au SVM linéaire standard et nettement meilleur que les arbres de classification. De plus, le la méthode proposée est robuste, c'est-à-dire qu'elle est stable en présence de valeurs aberrantes et invariante au changement d'échelle ou unités de mesure des variables prédictives. La méthode implique l'optimisation d'un linéaire Problème de programmation avec un grand nombre de variables de décision, pour lequel nous utilisons le célèbre

Ce travail a été partiellement soutenu par les projets MTM2005-09362-C03-01 de MEC, Espagne, et FQM-329 de Junta de Andaluc´ıa, Spain.

Technique de génération de colonnes.

Lorsque la règle de classification obtenue est trop complexe pour permettre une interprétabilité, une fonctionnalité wrapper

méthode de sélection est appliquée, donnant une règle de classification dont le comportement diffère légèrement de la règle de classification linéaire.

SVM et reste toujours meilleur que les arbres de classification.

Mots clés : Exploration de données, Classification supervisée, Génération de colonnes, Machines à vecteurs de support

1 Introduction et revue de la littérature

Classer des objets ou des individus en différentes classes ou groupes est l'un des objectifs du Data Mining.

Ce sujet a été abordé dans différents domaines tels que les statistiques, la recherche opérationnelle et l'artificiel.

Intelligence. Une introduction générale au Data Mining peut être trouvée dans [9].

Nous nous concentrons sur le problème bien connu de la classification supervisée, généralement appelé discrimination.

nant Analyse par des statisticiens, où l'on dispose d'un ensemble d'objets Ω et dont le but est de construire une règle de classification

qui prédit l'appartenance à la classe c u d'un objet u dans l'une d'un ensemble prédéfini de classes, au moyen de

son vecteur prédicteur x dans . Le vecteur prédicteur x u prend des valeurs dans un ensemble X qui est généralement supposé être un

sous-ensemble de l'IRp , tel que {0, 1} p , et les composantes x, = 1, 2, . . . , p, du vecteur prédicteur x sont appelés

variables prédictives. L'autre élément d'information définissant l'objet u, appelé appartenance à la classe c dans ,

prend des valeurs dans l'ensemble des classes C. L'objet u est dit appartenir à la classe c dans .

Toutes les informations sur les objets dans Ω ne sont pas disponibles, mais seulement dans un sous-ensemble l, appelé train-

échantillon, où le vecteur prédicteur et l'appartenance à la classe des objets sont connus. Avec ça

informations, la règle de classification doit être construite.

Au cours des quinze dernières années, les techniques de programmation mathématique ont été appliquées avec succès aux données.

Problèmes miniers, voir par exemple [2, 3, 16]. L'approche des machines à vecteurs de support (SVM) [4] est basée

sur la maximisation des marges, qui consiste à trouver l'hyperplan le plus éloigné du plus proche

objet. SVM s'est révélé être un outil très puissant pour la classification supervisée. Le plus populaire

Les versions de SVM intègrent, via une fonction du noyau, les variables prédictives d'origine dans une variable supérieure (éventuellement

infini) espace dimensionnel, [11]. De cette façon, on obtient des classificateurs avec de bonnes propriétés de généralisation

mais, en général, difficile à interpréter.

Dans certains domaines d'application, les praticiens, tels que les médecins ou les hommes d'affaires, peuvent être très réticents à utiliser

un classificateur qu'ils ne peuvent pas interpréter. Pour eux, les méthodes de Data Mining procèdent parfois comme une boîte noire,

ils ne se sentiraient donc pas suffisamment en confiance pour utiliser le classificateur à moins de pouvoir l'interpréter d'une manière ou d'une autre.

Par exemple, il est facile d'interpréter et de gérer des requêtes de type

- Est-ce une variable prédictive $_1$ gros ?
- Est-ce une variable prédictive $_2$ petit?
- Est-ce qu'une variable prédictive $_3$ atteindre une valeur très extrême ?

où les concepts de « grande », « petite » et « valeur extrême » doivent être quantifiés, par exemple sous la forme

la variable prédictive est-elle supérieure ou égale à b ?	(1)

Ce type de requêtes est utilisé par exemple dans les arbres de classification. De cette façon, les praticiens peuvent interpréter le cours.

sifier, décrivant son fonctionnement. De plus, ils peuvent voir directement quel rôle jouent les différentes variables prédictives

jouent dans le classificateur et détectent les valeurs d'une variable prédictive critique pour la classification.

Dans cet article, nous proposons un nouveau modèle qui, en utilisant des fonctions constantes par morceaux,

sélectionne l'échelle adéquate pour chaque variable. La règle obtenue est basée sur des requêtes de type (1), qui

rend l'interprétabilité plus facile. De plus, comme le montre empiriquement, les classificateurs obtenus sont robustes contre

la présence de valeurs aberrantes.

On se limite au cas où deux classes existent, $C = \{-1, 1\}$. Le cas multiclasse peut être

réduit à une série de problèmes à deux classes, comme cela a été suggéré par exemple dans [10, 11, 20].

Nous travaillons dans un framework basé sur SVM, où l'espace des fonctionnalités est défini en binarisant chaque variable,

et en considérant toutes les coupures possibles. Pour cette raison, nous appelons notre méthode le vecteur de support binarisé.

Machine, BSVM. Les résultats numériques montrent que l'approche proposée donne un classificateur qui se comporte de la même manière

au SVM linéaire standard et nettement meilleur que les arbres de classification. De plus, le compromis entre

l'interprétabilité et la capacité de classification peuvent être contrôlées via une limite supérieure sur le nombre de fonctionnalités autorisé.

Le classificateur proposé dans cet article, BSVM, est décrit dans la section 2. Puisque le nombre de fonctionnalités à considérer peut être énorme, la méthode BSVM génère un problème d'optimisation avec un grand nombre de variables de décision, à savoir (I) p , où (·) désigne la cardinalité d'un ensemble. Dans la section 3, une colonne-Un algorithme basé sur la génération est proposé afin de résoudre un tel problème d'optimisation. Résultats numériques sont présentés dans la section 4, tandis que les conclusions et quelques pistes de recherches futures sont discutées dans la section 5.

2 machines à vecteurs de support binarisés

Dans les applications pratiques, les règles simples de type (1) sont très souhaitables en raison de leur interprétabilité. Pour Par exemple, un médecin dirait que l'hypertension artérielle est un symptôme de maladie. Choisir le Le seuil b à partir duquel une pression artérielle spécifique serait considérée comme élevée n'est généralement pas une tâche facile.

On considère théoriquement toutes les règles possibles de type (1), formalisées mathématiquement par la fonction

$$\varphi_b(x) = \begin{cases} 1 & \text{si } x \geq b \\ 0 & \text{sinon} \end{cases} \quad (2)$$

pour b IR et = 1, 2, . . . , p. Dans ce qui suit, chaque variable prédictive est appelée variable, alors que chaque une fonction de type φb est appelée fonctionnalité.

Cette procédure de binarisation pourrait être effectuée lors d'une étape de prétraitement fastidieuse, où tous les éléments possibles des fonctionnalités sont créées. Cependant, nous ne le faisons pas comme une étape de prétraitement mais, comme nous le verrons plus tard, proposons une méthode pour générer des fonctionnalités en cas de besoin.

L'ensemble des valeurs seuils possibles b (et donc le nombre de caractéristiques) est, en principe, infini. Cependant, étant donné un échantillon d'apprentissage I, bon nombre de ces seuils possibles donneront exactement la même classification dans les objets en I, qui sont les objets dont les informations sont disponibles. En ce sens, pendant un certain temps variable prédictive et un échantillon d'apprentissage donné I, nous pouvons contraindre le choix de b IR à l'ensemble fini

B = {x u : u I}. De cette manière, la famille de fonctionnalités considérée est donnée par

$$F = \{\varphi b : b \quad B, = 1, 2, \ldots, p\}.$$

Ces caractéristiques permettent de classer de la manière suivante : à chaque caractéristique φb est associé un poids ωb, mesurer sa contribution pour le classement dans la classe −1 ou 1. La somme pondérée de ces caractéristiques et un seuil β constituent la fonction score,

$$f(x) = \omega\ \Phi(x) + \beta = \sum_{=1\ \{b \quad B|\ x \geq b\}}^{p} \omega b + \beta, \tag{3}$$

où Φ(x) = (φb(x))b B, =1,2,...,p et ω Φ(x) désigne le produit scalaire des vecteurs ω et Φ(x).

Les objets seront attribués à la classe −1 si f(x) < 0, et à la classe 1 si f(x) > 0. En cas d'égalité, c'est à dire f(x) = 0, les objets peuvent être alloués de manière aléatoire ou selon un ordre prédéfini. Dans cet article, suite à une Dans le pire des cas, ils seront considérés comme mal classés.

Pour une certaine variable prédictive , le coefficient ωb associé à la caractéristique φb représente le montant avec laquelle la requête « est-ce que x ≥ b ? » contribue à la fonction de score (3). Le poids ωb donne donc une connaissance approfondie de la façon dont la variable prédictive influence la classification, puisque nous remplaçons variable par la fonction s → $\{b \quad B|\ s \geq b\}\ \omega b$, déterminant ainsi le changement d'échelle à appliquer à variable. De plus, les variables pour lesquelles ωb sont nulles pour tout b B ne sont pas nécessaires pour le classification et peuvent être rejetés. Par exemple, en prenant la base de données du Wisconsin sur le cancer du sein de le référentiel UCI Machine Learning, [14], avec des données de diagnostic de cancer, et en utilisant le BSVM, il il s'avère que seules 12 variables sur 30 ont au moins un ωb non nul. Autrement dit, seulement 12 sur les 30 variables sont pertinentes pour la classification. Dans la figure 1 nous montrons, pour chacune de ces douze variables, sa contribution à la fonction de score. A titre d'illustration, le tableau 1 montre, pour la variable Pire Texture, sa seuils b, les poids correspondants ωb et les poids cumulés $b \leq b\ \omega b$.

Nous avons également tracé sur la figure 1 la médiane (représentée par une étoile) et la moyenne (représentée par un croix). On peut voir comment, même si la moyenne, ou la médiane, constitue parfois un bon choix pour le seuil, cela n'arrive pas en général et BSVM préfère d'autres choix.

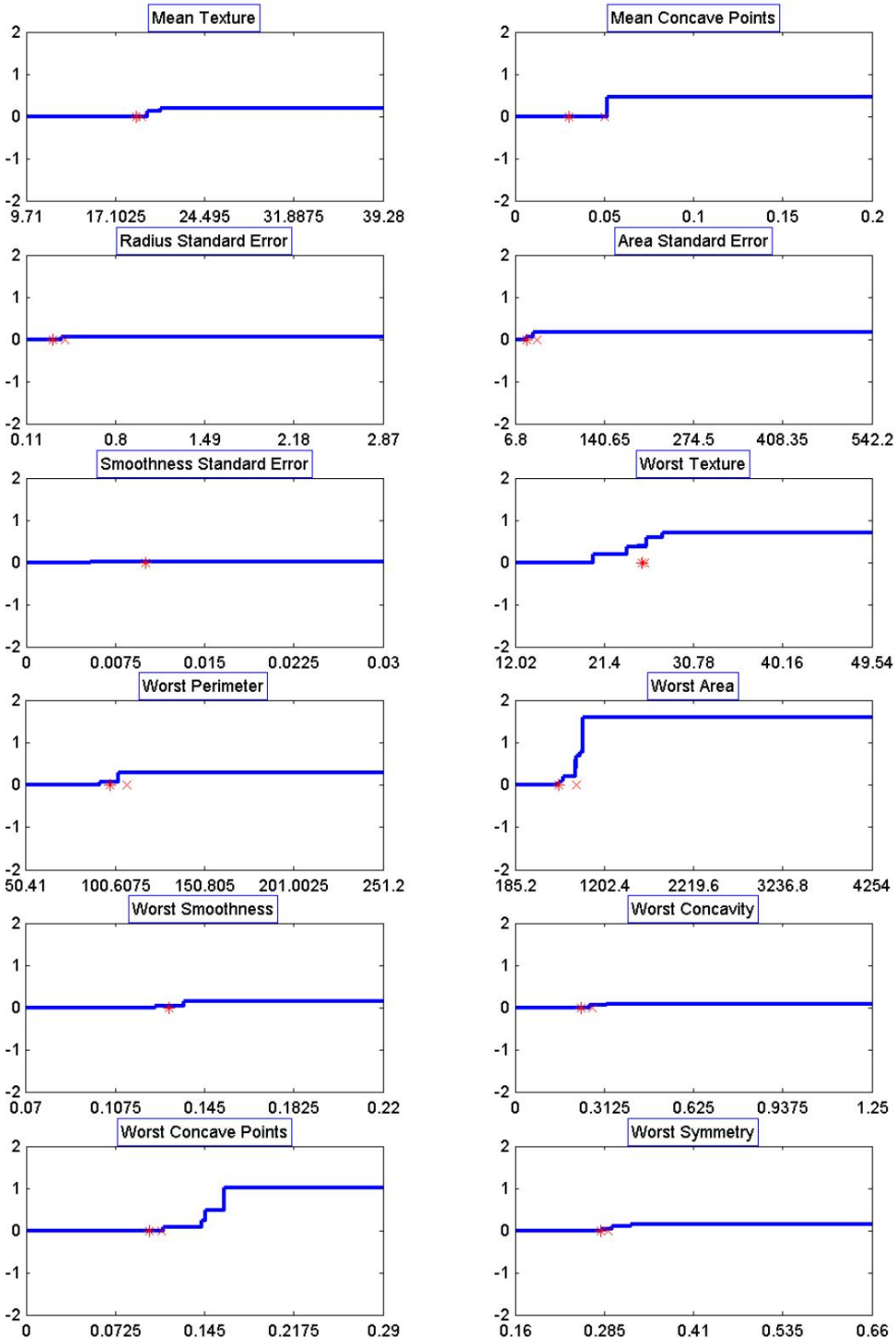

Figure 1 : Choix automatique des échelles

b	ωb	b ≤b ωb
20,24	0,17795	0,17795
23,75	0,19729	0,37524
25,73	0,01160	0,38685
25,84	0,20116	0,58801
27,57	0,11219	0,70019

Tableau 1 : Échelle pour la variable Pire texture.

Étant donné que le résultat des fonctionnalités proposées dans cet article est toujours binaire, l'importance représentée par le coefficient est toujours mesuré dans la même échelle. Le praticien pourrait choisir d'interpréter ces caractéristiques avec le coefficient le plus élevé en valeur absolue, obtenant ainsi une grande compréhension du comportement du classificateur. Toutefois, si le praticien préfère maintenir un nombre de fonctionnalités faible afin d'obtenir une image complète du fonctionnement du classificateur, puis une procédure de sélection des fonctionnalités du wrapper, comme dans [7], pourrait être utilisé pour gagner en interprétabilité. Dans la section 4.3, nous donnons quelques résultats numériques en utilisant une méthode simple mais une puissante procédure d'encapsulation connue sous le nom d'élimination de caractéristiques récursives, comme proposée pour la première fois dans [8].

Afin de choisir ω et β nous suivons une approche basée sur SVM qui consiste à trouver l'hyperplan qui maximise la marge dans l'espace des caractéristiques, c'est à dire l'espace des images Φ(x u) des objets u dans l'échantillon d'apprentissage I. L'utilisation de la maximisation de la marge est théoriquement motivée par les limites de la capacité de généralisation [17, 19, 20], où la probabilité de mal classifier un prochain individu est borné par une fonction décroissante dans la marge. L'approche SVM dite à marge dure propose le choix de l'hyperplan à marge maximale, c'est à dire l'hyperplan qui classe correctement tous les objets dans I et est le plus éloigné de l'objet le plus proche. En revanche, l'approche dite de la marge souple, permet à certains objets d'être mal classés. Nous utilisons cette dernière version car elle a été démontrée empiriquement éviter le surajustement, un phénomène qui se produit lorsqu'un faible taux d'erreurs de classification en I ne se généralise pas

aux objets à venir. Le problème de maximisation de la marge souple est formulé dans cet article par

$$\begin{aligned} \min \quad & o + C \sum_{u=1}^{l} \xi_u \\ \text{st : c} \quad & \omega \Phi(x_u) + \beta + \xi_u \geq 1 \qquad u \in I \\ & \xi_u \geq 0 \qquad u \in I \\ & \omega \in \mathbb{R}^N, \beta \in \mathbb{R}, \end{aligned} \tag{4}$$

où les variables de décision sont le vecteur de poids ω, la valeur seuil β et les perturbations ξ u associées avec la mauvaise classification de l'objet u I. · désigne la norme L1 , $N = \sum_{=1}^{p}$ (B) et C est une constante qui compromet la marge des objets correctement classés et les perturbations ξ u . Un approprié la valeur de C est généralement choisie par des techniques de validation croisée, voir par exemple [12].

Alors que la distance entre les points a généralement été considérée dans la littérature comme la distance euclidienne norme, donnant la marge à mesurer également par la norme euclidienne, d'autres normes telles que la L1 et la norme L∞ ont été considérées, [13], et ont été appliquées avec succès, voir par exemple [1, 18, 21].

Contrairement au cas euclidien, dans lequel un hyperplan de marge maximale peut être trouvé en résolvant un programme quadratique avec contraintes linéaires, dans le cas de la norme L1 , utilisé dans cet article, une solution optimale peut être trouvé en résolvant un problème de programmation linéaire (LP). Dans [15], les résultats empiriques montrent que « dans en termes de performances de séparation, L1, L∞ et le SVM basé sur la norme euclidienne ont tendance à être assez similaires. De plus, les normes polyédriques, comme la norme L1 , contribuent à la parcimonie du classificateur, donnant ω avec de nombreux composantes égales à zéro, voir par exemple [5].

Puisque nous utilisons la norme L1 , le problème (4) peut être formulé comme le problème LP suivant,

$$\begin{aligned} \min \quad & \sum_{=1b}^{p} B \; (\omega - b)^{+} + C + \omega b \sum_{u=1}^{l} \xi_u \\ \text{St :} \quad & \sum_{=1\,b}^{p} B \; (Oh^{+} - \omega b - b)c_u \varphi b(x_u) + \beta c_u + \xi_u \geq 1 \qquad u \in I \\ & oh_b^{+} \geq 0 \qquad b \in B, \; = 1, 2, \ldots, p \\ & \text{[illegible]}^{-} \geq 0 \qquad b \in B, \; = 1, 2, \ldots, p \\ & \xi_u \geq 0 \qquad u \in I \\ & \beta \in \mathbb{R}. \end{aligned} \tag{5}$$

Après avoir trouvé l'hyperplan de marge maximale dans l'espace des caractéristiques défini par F, la fonction de score a la forme décrite en (3).

Pour chaque caractéristique, la valeur absolue de son coefficient indique l'importance de cette caractéristique pour le classification. En utilisant la théorie de base de la programmation linéaire, il est facile de voir que le nombre de fonctionnalités avec le coefficient non nul n'est pas supérieur au nombre d'objets dans la base de données. Toutefois, ce numéro peut être toujours important pour l'interprétabilité du classificateur.

Génération à 3 colonnes

Dans cet article, nous proposons que le problème (5) soit résolu par l'outil de programmation mathématique bien connu. appelé Column Generation, initialement introduit pour le problème du matériel de coupe [6]. Au lieu de résoudre directement le problème (5), qui comporte un nombre élevé de variables de décision, la technique de génération de colonnes résout une série de problèmes réduits où les variables de décision, correspondant aux caractéristiques de l'ensemble F, sont ajouté de manière itérative selon les besoins.

Pour F F, soit le problème principal (5-F) le problème (5) avec la famille de fonctionnalités F. À chaque itération, nous résolvons le problème (5-F). L'étape suivante consiste à vérifier si la solution actuelle est optimale pour le problème. (5) ou non, et, dans ce dernier cas, générer une nouvelle caractéristique φ améliorant la valeur objective du courant solution. La fonctionnalité générée est ajoutée à la famille de fonctionnalités F. Ce processus est répété jusqu'à ce que l'optimalité est atteinte.

Afin de générer de nouvelles fonctionnalités, la technique de génération de colonnes utilise la double formulation de Problème (5),

$$\begin{aligned} \text{maximum} \quad & \sum_{u \in I} \lambda_u \\ \text{st} : -1 \le & \sum_{u \in I} \lambda_u c_u \varphi(x_u) \le 1 \quad \varphi \in F \\ & \sum_{u \in I} \lambda_u c_u = 0 \\ & 0 \le \lambda_u \le C \quad u \in I. \end{aligned} \tag{6}$$

La formulation double du problème principal (5-F) ne diffère de celle-ci que par le premier ensemble de contraintes, qui devrait être atteint φ F au lieu de φ F.

Soit (ω , β) une solution optimale du problème principal (5-F), et soit (λ)u I les valeurs du

solution double optimale correspondante. Si la solution optimale du problème principal (5-F) est également optimale

pour le problème (5), alors, pour chaque caractéristique φ F, les contraintes du double problème (6) seront valables, c'est-à-dire

$$-1 \leq \sum_{u \in I} je^{*}_{\text{dans}} c\, u\varphi(x\, u) \leq 1. \qquad (7)$$

Notons Γ(φ) = $\sum_{u \in I} \lambda_{\text{dans}} c\, u\varphi(x\, u)$. Si (ω , β) n'est pas optimal pour le problème (5), alors la méthode la plus violée

la contrainte nous donne des informations sur quelle fonctionnalité est prometteuse et pourrait être ajoutée à F, dans le sens

que l'ajout d'une telle fonctionnalité à l'ensemble F donnerait, à cette itération, la plus forte amélioration du

fonction objectif. Ainsi, nous souhaitons générer une nouvelle caractéristique φ F maximisant |Γ(φ)|. Trouver un tel φ

peut se réduire à résoudre deux problèmes d'optimisation :

- maxφ F Γ(φ),
- minφ F Γ(φ).

Dans la suite de cette section, nous donnons une description plus détaillée de la mise en œuvre du

Algorithme de génération de colonnes.

3.1 Ensemble initial de fonctionnalités

Dans la procédure de génération de colonnes, de nouvelles fonctionnalités sont ajoutées séquentiellement au problème, en fonction des

valeurs doubles de la solution actuelle. La technique de génération de colonnes commence par un premier Master

Problème, c'est à dire qu'un ensemble de fonctionnalités F0 doit être choisi pour initialiser l'algorithme. Nous avons choisi de commencer

avec une caractéristique par variable, avec le seuil fixé égal à sa médiane dans les objets de I.

3.2 Génération de fonctionnalités

Trouver le meilleur φ F se réduit à trouver une variable prédictive et un seuil b B, tels que |Γ(φb)|,

avec φb défini par (2), est maximal. Dans cette section, nous décrivons un algorithme pour corriger le prédicteur

variable , trouver le seuil b maximisant Γ(φb). Trouver le minimum peut être fait de la même manière.

Tout d'abord, nous trions tous les objets par ordre décroissant selon les valeurs de la variable prédictive. Noter par u(i) l'objet en i-ième position. Pour simplifier, supposons qu'il n'y ait pas de valeurs répétées, c'est-à-dire $x_{u(1)} > x_{u(2)} > \ldots > x_{u((l))}$. Le cas des valeurs répétées sera analysé plus tard.

Une fois est corrigé et tous les objets sont triés par les valeurs de la variable prédictive , la valeur Γ(φb) peut être calculé efficacement avec une procédure récursive. En effet, avec certitude $i \in \{1, 2, \ldots, (l)\}$, nous avons $\Gamma(\varphi b_{i+1}) = \Gamma(\varphi b_i) + \lambda_{u(i+1)} c_{u(i+1)}$, où b_i désigne le seuil choisi comme x . De plus, puisque $\lambda_{u(i)} = \lambda_{dans}$ est tout $u \in I$, chaque fois que $c_{u(i+1)} = 1$, alors $\Gamma(\varphi b_{i+1}) > \Gamma(\varphi b_i)$. Ainsi, vérifier si φb_i est non négatif pour maximum n'est pas nécessaire pour chaque i, mais seulement pour ceux i tels que $c_{u(i)} = 1$ et $c_{u(i+1)} = -1$.

Dans le cas où il y a des valeurs répétées dans $\{x_u : u \in I\}$, la règle ci-dessus ne s'applique pas. Laissez-moi et t soit tel que $x_{u(i-1)} > x_{u(i)} = x_{dans(i+1)} = \ldots = x_{u(je+t)} > x_{u(i+t+1)}$. Notez que, dans l'ensemble des objets où La variable prédictive a la même valeur, il peut y avoir des objets appartenant à des classes différentes. Dans ce cas, être vérifié, quelle que soit la valeur de c . Cependant, si $c_{u(i+t+1)} = c_{u(i)}$ alors $b = b_i$ doit $= c_{dans(i+1)} = \ldots = c_{tu(i+t)}$ et $c_{u(i+t+1)} = 1$ nous savons que le réglage $b = b_j$, pour tout $j = i, i + 1, \ldots, i + t$, sera amélioré par définissant $b = b_{i+t+1}$. Cela signifie que $b = b_i$ ne donne pas un maximum de Γ(φb). Seulement si $c_{u(i)} = 1$ et $c_{u(i+t+1)} = -1$, il vaut la peine de considérer $b = b_i$ comme candidat pour être le maximum.

La minimisation de Γ se fait de manière analogue. Par exemple, en cas d'absence de valeurs répétées, les candidats à être un minimum correspondre aux objets u(i) appartenant à la classe −1 à laquelle appartient l'objet suivant u(i + 1) à la classe 1.

En tenant compte de toutes ces considérations, nous obtenons, pour une variable prédictive fixe , Compte tenu du valeurs doubles λ_{dans} . l'algorithme décrit ci-dessous, qui trouve le seuil b^+ (et respectivement, b^-) pour lequel $\Gamma(\varphi b^+)$ (respectivement $\Gamma(\varphi b^-)$) est maximal (respectivement minimal).

Algorithme 1 : Choisir un seuil pour la variable.

Étape 0. Triez les objets par x : x_{dans} .

Étape 1. Définir $i \leftarrow 1$, somme$\leftarrow 0$, max$\leftarrow 0$ et min$\leftarrow 0$.

Étape 2. Définir somme$\leftarrow$ somme$+\lambda_{dans} c_{u(i)}$.

Étape 3. Étape 3.0. Si x , alors passez à l'étape 4. = x

3.1. Sinon, si pour certains t > 0, x existe j avec j = 1, . . . , t et c , alors:

- Si somme>max, alors définissez max ← somme.
- Si somme <min, alors définissez min ← somme.

Étape 3.2. Sinon,

- si c = 1, cu(i+1) = −1 et somme>max, puis définissez max ← somme.
- si c = −1, cu(i+1) = 1 et somme<min, puis définissez min ← somme.

Étape 4. Définissez i ← i + 1. Si i ≤ (I), passez à l'étape 2, sinon STOP.

3.3 Détails de mise en œuvre

L'algorithme de génération de colonnes a été implémenté comme suit. L'ensemble initial de fonctionnalités F0 est construit, comme décrit à la section 3.1, en utilisant des caractéristiques dont les seuils sont les médianes des variables prédictives. Alors, Le problème (5-F0) est résolu pour un tel ensemble initial de fonctionnalités. Les valeurs duales de la solution optimale trouvée, sont utilisés pour générer de nouvelles fonctionnalités.

À chaque étape de l'algorithme de génération de colonnes, au lieu de générer une seule fonctionnalité (celle en maximisant |Γ(φ)|), nous générons deux caractéristiques pour chaque variable prédictive , donné par les seuils pour lesquels Γ(φb) est maximal et minimal. Cela se fait en utilisant l'algorithme 1, comme décrit dans la section 3.2. Nous le faisons pour toutes les variables prédictives, obtenant ainsi des fonctionnalités 2p. Ces fonctionnalités générées ayant |Γ(φ)| > 1 sont ajouté à F et le problème LP (5-F) est résolu. Ces étapes sont répétées jusqu'à ce que toutes les fonctionnalités générées avoir |Γ(φ)| ≤ 1, auquel cas nous avons trouvé une solution optimale du problème (5). Un résumé de ceci L'algorithme de génération de colonnes est décrit comme suit.

Algorithme de génération de colonnes.

Étape 0. Fixez F0 ← {φ1b , φ2b , . . . , φpb }, où b est la médiane de la variable prédictive , pour = 1, 2, . . . , p. Réglez F ← F0.

Étape 1. Résoudre le problème (5-F). Soit (ω , β) sa solution optimale, de valeurs duales λ dans . u l.

Étape 2. Pour chacun = 1, 2, . . . , p fais :

Étape 2.0. Exécutez l'algorithme 1 pour choisir b + .

Étape 2.1. Si Γ(φb+) > 1, alors posez F ← F {φb+ }.

Étape 2.2. Exécutez l'algorithme 1 pour choisir b - .

Étape 2.3. Si Γ(φb−) < −1, alors posez F ← F {φb− }.

Étape 3. Si F a été modifié, alors passez à l'étape 1, sinon STOP : nous avons trouvé une solution optimale

du problème (5).

4 Résultats numériques

4.1 Bases de données

Nous allons d'abord analyser la capacité de classification de l'ensemble des fonctionnalités proposées dans l'article.

Dans ce but, une série d'expériences numériques ont été réalisées à l'aide de bases de données accessibles au public.

du référentiel d'apprentissage automatique UCI [14]. Un résumé des caractéristiques des bases de données

utilisé dans les expériences est présenté dans le tableau 2. Cinq bases de données différentes ont été utilisées, à savoir le Cylindre

Base de données de bandes, appelées ici bandes ; les bases de données de filtrage du crédit, appelées ici crédit ; l'ionosphère

Base de données, appelée ici ionosphère ; la base de données Sonar, appelée ici sonar ; et le nouveau diagnostic

Base de données contenue dans les bases de données sur le cancer du sein du Wisconsin, appelée ici wdbc.

Pour chaque base de données, le nom du fichier (tel qu'appelé dans la base de données), le nombre de variables prédictives

(tous quantitatifs) p et le nombre total d'objets (Ω) sont donnés dans le tableau 2. En cas d'existence de

valeurs manquantes, comme cela se produit dans les tranches et le crédit, les objets avec des valeurs manquantes ont été supprimés du

base de données. Dans les bases de données sur les tranches et le crédit, certaines des variables prédictives étaient nominales. Chacun de ces

Les variables prédictives ont été remplacées par un ensemble de variables binaires de la manière suivante : pour chaque

valeur ˆx de la variable nominale d'origine , une nouvelle variable binaire est construite en prenant la valeur un lorsque x est égal à ˆx et zéro sinon.

nom	nom de fichier p (Ω)
bandes	bands.data 56 365 crx.data
crédit	43 666 ionosphere
ionosphere.data 34 351 sonar.all-data 60 208	
sonar	wdbc.data 30 569
wdbc	

Tableau 2 : Informations sur les bases de données.

Afin de comparer la qualité du classificateur BSVM avec la qualité de classification d'autres classificateurs, nous avons testé les performances de deux méthodes de référence très différentes : les arbres de classification, tous deux avec élagage (prTree) et sans élagage (Tree), et SVM avec noyau linéaire. Tous les résultats présentés sont obtenu par validation croisée 10 fois, par exemple [12]. Les pourcentages moyens d'objets correctement classés dans l'échantillon d'apprentissage (tr) et l'échantillon de test (test) sont affichés dans le tableau 3 pour différentes valeurs de le paramètre C. Trouver une procédure pour dériver un choix approprié de la valeur C, qui compromet la marge et les écarts des objets mal classés, dépassent le but de cet article. CPLEX 8.1.0 a été utilisé comme solveur LP.

C'est un fait bien connu en SVM que, si le paramètre C est choisi trop proche de zéro, on peut obtenir comme solution optimale du problème (5) un vecteur avec ω = 0, à partir duquel un classificateur trivial attribuant tous les objets à une classe est obtenu. Cette situation dégénérée (indiquée dans les tableaux 4 à 5 par dc) est évitée en prenant un plus gros C.

4.2 Comparaison avec d'autres techniques

Les résultats du BSVM sont présentés dans le tableau 4, où les pourcentages moyens d'objets correctement classés dans les échantillons de formation et de test sont affichés avec le nombre de fonctionnalités générées (fonctionnalités) avec un coefficient non nul dans le classificateur, et le nombre de variables prédictives réellement utilisées par le classificateur (var). Comme le montrent les résultats, le BSVM se comporte considérablement mieux que les arbres de classification. et comparable à la technique SVM linéaire standard. Les arbres de classification sont largement utilisés dans les applications

des domaines aussi divers que la médecine (diagnostic), l'informatique (structures de données), la botanique (classification) et psychologie (théorie de la décision), principalement parce qu'ils sont faciles à interpréter. Nous affirmons que le BSVM conserve cette propriété sans perdre la bonne capacité de classification du SVM linéaire standard.

4.3 Réduire le nombre de fonctionnalités

Les résultats du tableau 4 montrent que le nombre d'entités est généralement supérieur à une, voire deux centaines, ce qui rend difficile la détection des fonctionnalités les plus pertinentes. Afin d'obtenir un classificateur plus interprétable, nous procéder à une procédure d'élagage dans laquelle les fonctionnalités sont supprimées de manière récursive. Dans cette procédure, qui a été appliqué avec succès dans SVM standard, voir [8], toutes les caractéristiques générées avec un coefficient nul dans le classificateur et la caractéristique avec un coefficient non nul ayant la plus petite valeur absolue sont éliminés. Ensuite, les coefficients sont recalculés par l'optimisation du problème LP (5). Cette élimination La procédure est répétée jusqu'à ce que le nombre de fonctionnalités soit inférieur à un nombre donné à l'avance.

Les pourcentages moyens d'objets correctement classés dans l'échantillon de formation et de test sont affichés, dans le tableau 5, lorsque la procédure d'élimination est appliquée jusqu'à ce qu'il reste 30 éléments ou moins dans la classification règle. Comme on peut le constater, la capacité de classification se détériore légèrement, mais reste toujours meilleure que celui des Arbres de Classification.

4.4 Comportement en présence de valeurs aberrantes

Le classificateur proposé dans cet article est basé sur des fonctions de seuil, il semble donc que des observations extrêmes Les valeurs très élevées ou très faibles n'auront pas une forte influence sur le classificateur. De manière empirique Pour tester ce fait, une série d'expériences ont été réalisées dans lesquelles certaines valeurs aberrantes ont été artificiellement introduites. dans la base de données. Chaque cellule de la base de données wdbc a été choisie comme valeur aberrante avec une probabilité de 0,05. Ceux Les cellules choisies ont été modifiées en ajoutant ρ fois la plage de sa variable prédictive, pour ρ = 10, 100, 1000. Dans Les tableaux 6 et 7 des résultats de la base de données wdbc sont présentés pour les méthodes de référence et pour le BSVM. Le La capacité de classification du classificateur linéaire SVM se détériore considérablement lors de l'introduction de valeurs aberrantes, alors que le BSVM n'est pratiquement pas affecté.

5 Conclusions et recherches complémentaires

Dans cet article, un nouvel outil basé sur SVM pour la classification supervisée a été proposé où le classificateur
donne des connaissances approfondies sur la manière dont les variables prédictives influencent la classification. En effet, le
le comportement de non-linéarité des données est modélisé par le classificateur BSVM à l'aide de requêtes simples, de type (1),
facilement interprétable par les praticiens, par exemple par des représentations similaires à la figure 1. Sa classification
Le comportement, qui se situe entre SVM et les arbres de classification, fait de BSVM un outil intéressant lorsqu'un bon
Une capacité de classification est requise, mais l'interprétabilité des résultats est une question importante. Notre numérique
l'expérience montre que BSVM est beaucoup plus robuste que SVM contre les valeurs aberrantes.

La procédure de binarisation a été appliquée à chaque variable séparément. Si les interactions entre
On s'attend à ce que les variables soient pertinentes, des procédures de binarisation plus générales pourraient être envisagées. Ce
La question sera abordée dans un prochain article.

Les références

[1] K. Bennet. Combinaison de méthodes de vecteurs de support et de programmation mathématique pour l'induction. Dans
B. Scholkopf, C. Burges et A. Smola, éditeurs, Advances in Kernel Methods: Support Vector Learn-
ing, pages 307 à 326. Presses du MIT, 1999.

[2] PS Bradley, OL Mangasarian et D. Musicant. Méthodes d'optimisation dans des ensembles de données massifs. Dans
J. Abello, PM Pardalos et MGC Resende, éditeurs, Handbook of Massive Datasets, pages 439–
471. Kluwer Academic Publishers, Boston, 2002.

[3] E. Carrizosa et B. Mart´ın-Barrag´an. Classification en deux groupes via une maximisation de la marge bi-objectif.
modèle de tion. Journal européen de recherche opérationnelle. Dans la presse.

[4] C. Cortés et V. Vapnik. Réseaux de vecteurs de support. Apprentissage automatique, 20(3):273-297, 1995.

[5] G. Fung et OL Mangasarian. Une méthode de Newton de sélection de fonctionnalités pour une machine à vecteurs de support
classification. Optimisation informatique et applications, 28(2):185-202, 2004.

[6] PC Gilmore et RE Gomory. Une approche de programmation linéaire du problème du stock de coupe.
Recherche opérationnelle, 9 : 849-859, 1961.

[7] I. Guyon et A. Elisseff. Une introduction à la sélection de variables et de fonctionnalités. Journal de la machine
Learning Research, 3 : 1157-1182, mars 2003.

[8] I. Guyon, J. Weston, S. Barnhill et V. Vapnik. Sélection de gènes pour la classification du cancer à l'aide
prendre en charge les machines vectorielles. Apprentissage automatique, 46 : 389-422, 2002.

[9] H. Hand, H. Mannila et P. Smyth. Principes de l'exploration de données. Presses du MIT, 2001.

[10] T. Hastie et R. Tibshirani. Classification par couplage par paires. Les Annales de la statistique, 26(2):451–
471, 1998.

[11] R. Herbrich. Apprentissage des classificateurs de noyau. Théorie et algorithmes. Presses du MIT, 2002.

[12] R. Kohavi. Une étude de validation croisée et de bootstrap pour l'estimation de la précision et la sélection de modèles.
Dans Actes de la 14e Conférence internationale conjointe sur l'intelligence artificielle, pages 1137-1143.
Morgan Kaufman, 1995.

[13] OL Mangasarien. Séparation linéaire et non linéaire des motifs par programmation linéaire. Opérations
Recherche, 13 : 444-452, 1965.

[14] DJ Newman, S. Hettich, CL Blake et CJ Merz. Référentiel UCI d'apprentissage automatique
Bases de données. http://www.ics.uci.edu/ mlearn/MLRepository.html, 1998. Université de Californie,
Irvine, Département de l'information et des sciences informatiques.

[15] JP Pedroso et N. Murata. Machines vectorielles de support avec différentes normes : motivation, formulations
et les résultats. Lettres de reconnaissance de formes, 22 : 1263-1272, 2001.

[16] AM Rubinov, AM Bagirovand, NV Soukhoroukova et J. Yearwood. Sans surveillance et super-
classification des données via une optimisation non fluide et globale. HAUT, 11(1):1-93, 2003.

[17] J. Shawe-Taylor, PL Bartlett, RC Williamson et M. Anthony. Minimisation des risques structurels sur
hiérarchies dépendantes des données. Transactions IEEE sur la théorie de l'information, 44(5):1926-1940, 1998.

[18] A. Smola, TT Friess et B. Schölkopf. Support semi-paramétrique programmation vectorielle et linéaire
Machines. Dans MJ Kearns, SA Solla et DA Cohn, éditeurs, Advances in Neural Information
Systèmes de traitement 10, pages 585 à 591. Presses du MIT, 1999.

[19] V. Vapnik. La nature de la théorie de l'apprentissage statistique. Springer-Verlag, 1995.

[20] V. Vapnik. Théorie de l'apprentissage statistique. Wiley, 1998.

[21] J. Weston, A. Gammerman, MO Stitson, V. Vapnik, V. Vovk et C. Watkins. Vecteur de soutien
estimation de la densité. Dans B. Schölkopf, C. Burges et A. Smola, éditeurs, Advances in Kernel Methods
- Apprentissage des vecteurs de support, pages 293 à 305. MIT Press, 1999.

bandes			
méthode C	% tr	% essai	
SVM 0,01	SVM	64.44	64.44
0,10	SVM 1	74.57	65,93
SVM 10	81,65	80.16	71,85
SVM 100	82,18	SVM 1000	72,96
82,35	prTree	Arbre	72.22
			72.59
		74.12	64,81
		92.43	66,67
crédit			
méthode C	% tr	% essai	
SVM 0,01		86.36	86.31
SVM 0,10		86.31	86.31
SVM 1	SVM 10	86,74	85,85
86,80	SVM 100	86,91	SVM
1000	87,09	prTree	Arbre
			85,85
			85.54
		86.31	86.31
		95.15	81,69
ionosphère			
méthode C	% tr	% essai	
SVM 0,01		66.25	65,71
SVM 0,10		89.21	87,71
SVM	1	91,84	87,71
SVM10		94.03	88.29
SVM100	95,21		87,71
SVM1000	95,65		86.29
prArbre		91.17	89.43
Arbre		98.03	87,71
sonar			
méthode C	% tr	% essai	
SVM 0,01		54.56	54h00
SVM 0,10		84.33	75.00
SVM	1	88.39	74.50
SVM10		92.28	73.50
SVM100	97.06		75.00
SVM 1000	100,00		73.50
prArbre		82,56	71.50
Arbre		97,56	77.00
wdbc			
méthode C	% tr	% essai	
SVM 0,01	87,16	86,79	
SVM 0,10	95,95	95,71	
SVM1	98,27	98,04	
SVM10	98,45	97,68	
SVM 100	99,11	96,96	
SVM1000	99,50		96.43
prArbre		96.71	94.46
Arbre		99.31	93,75

Tableau 3 : Comportement de classification dans les méthodes de référence.

bandes				
C %	tr %	essai	caractéristiques	était
0,01		cc		
0,1	cc			
0,1	98,85	10 73,33	121.1	28,0
100,00	100	72,59	128,8	28.3
100,00	1000	72,22	128,5	28.3
100,00		72,59	128,4	28.4
crédit				
C %	tr %	essai 0,01 86,31	caractéristiques	était
86,31	0,1	86,31 86,31 0,1	1,0	1.0
95,71	82,92		1,0	1.0
			138,2	21.7
10	100,00	80.00	199,5	24,8
100	100,00	79,69	200,3	24,7
1000	100,00	80.31	200,5	24,8
ionosphère				
C %	tr %	fonctionnalités de test		était
0,01		cc		
0,1	91.17	90.57	2.0	2.0
0,1	100,00	90.57	92,7	31.1
10	100,00	90.57	93.1	31.2
100	100,00	90.57	93.1	31.2
1000	100,00	90.57	93,4	31.1
sonar				
C %	tr %	essai	caractéristiques	était
0,01		cc		
0,1	91,83	75.00	39.2	25.9
0,1	100,00	80,50	97,5	47,8
10	100,00	80.00	97,4	47,8
100	100,00	80,50	97,5	47,8
1000	100,00	80,50	97,5	47,8
wdbc				
C %	tr %	essai	caractéristiques	était
0,01	92,60	90.54	1.0	1.0
0,1	97.74	96.07	21.1	9.0
0,1	100,00	95.71	68,4	24,8
10	100,00	96.25	68,2	25,0
100	100,00	95,89	67,6	25,0
1000	100,00	96.07	68,0	25,0

Tableau 4 : Comportement de classification sur BSVM.

BSVM (réduction du nombre de fonctionnalités)										
	bandes		crédit		ionosphère		sonar		wdbc	
C	% tr	% test	% tr	% test	% tr	% test	% tr	% test	% tr	% test
0,01	cc		86.31	86.31	cc		cc		92,60	90.54
0,1	cc		86.31	86.31	91.17	90.57	91,94	76.50	97.74	95,89
1	92.14	72,22	91,93	84,00	100,00	90,00	100,00	77,50	100,00	95.71
10	94,81	66.30	89.50	80,92	100,00	89.43	100.00	77,00	100,00	96.07
100	95,47	66h30	90h09	82,00	100,00	89,71	100,00	77,00	100,00	96.07
1000	95.14	66,67	91,42	80,77	100,00	89.43	100.00	77,00	100,00	96.25

Tableau 5 : Réduire le nombre de fonctionnalités à 30 au maximum.

Méthodes de référence avec valeurs aberrantes. Base de données wdbc										
		original		ρ = 1		ρ = 10		ρ = 100		
méthode	C	% tr	% test	% tr	% test	% tr	% test	% tr	% test	
SVM	0,01	87,16	SVM	0,10	86,79	65,26	65,18	63,21	63,21	63,21
95,95	SVM	1	98,27	SVM	95,71	90,24	88,75	64,13	63,21	63,47
62,86	10	98,45	SVM	100	99,11	98.04	91.53	90,18	65,65	59,64
64,94	62,86	SVM	1000	99,50	prTree	97,68	92,24	88,57	64,54	57,68
64,96	61.61	Arbre				96,96	92,42	88,57	64,72	57,50
64,94	60,89					96,43	92,40	88,75	64,70	57,50
64,94	60,71				96.71	94,46	96,31	93.04	95.71	92,14
95,99	92.32				99.31	93,75	98,95	93,75	98,95	93,75
98,95	93,75									

Tableau 6 : Comportement de classification des méthodes de référence avec valeurs aberrantes.

BSVM avec valeurs aberrantes. Base de données wdbc								
	original		ρ = 1		ρ = 10		ρ = 100	
C	% tr	% test	% tr	% test	% tr	% test	% tr	% test
0,01	92,60	90.54	90.00	85,89	90.00	85,89	90.00	85,89
0,1	97.74	96.07	98.35	95.36	98.35	95.36	98.35	95.36
1	100,00	95,71	100,00	95.18	100.00	95.18	100.00	95.18
10	100,00	96.25	100.00	95,00	100,00	95,00	100,00	95.00
100	100,00	95,89	100,00	95,00	100,00	95,00	100,00	95.00
1000	100,00	96,07	100,00	95,18	100,00	95,18	100,00	95.18

Tableau 7 : Comportement de classification de BSVM avec valeurs aberrantes.