# Application des noyaux multiples de type *Kernel Basis* à la méthode *Relevance Vector Machine* pour la sélection de modèles

Frédéric Suard, David Mercier

CEA, LIST, Laboratoire Intelligence Multi-capteurs et Apprentissage,
F-91191 Gif sur Yvette, FRANCE.
prenom.nom@cea.fr

**Résumé** − Nous présentons ici une adaptation des noyaux multiples de type *Kernel Basis* à l'algorithme Relevance Vector Machine. L'intérêt du *Kernel Basis* réside dans la capacité d'adapter des noyaux globaux et locaux dans une même solution. La finalité consiste en effet à affecter aux vecteurs de la solution un ensemble de noyaux qui soit spécifique à chaque vecteur. Nous proposons d'utiliser cette approche pour un problème de sélection de variables, afin de choisir pour chaque vecteur les variables les plus pertinentes. Les performances obtenues sur les résultats préliminaires et comparées avec une approche de noyau multiple composite, sont très prometteuses et ouvrent de nouvelles perspectives.

**Abstract** − This paper presents an extension of multiple kernels like Kernel Basis to the Relevance Vector Machine algorithm. The framework of kernel machines has been a source of many works concerning the merge of various kernels to build the solution. Within these approaches, Kernel Basis is able to combine both local and global kernels. The interest of such approach resides in the ability to deal with a large kind of tasks in the field of model selection, for example the feature selection. We propose here an application of RVM-KB to a feature selection problem, for which all data are decomposed into a set of kernels so that all points of the learning set correspond to a single feature of one data. The final result is the selection of the main features through the relevance vectors selection.

## 1 Introduction

Depuis de nombreuses années les méthodes à noyaux ont démontré leur fort potentiel en discrimination ou régression, comme par exemple le classifieur *Support Vector Machine* [10], *kernel PCA* [8] ou bien encore *kernel Fisher discriminant* [6]. L'intérêt de la plupart de ces approches réside dans la définition de la fonction de prédiction à l'aide d'une combinaison linéaire pondérée de fonctions noyaux. La méthode *Support Vector Machine*, en particulier, est applicable sur de nombreux problèmes, avec une grande efficacité, grâce aux différentes adaptations proposées telles que le *one-class*, la régression, l'utilisation d'un noyau composite [1]. Néanmoins, les solutions définies ont l'inconvénient de ne pas être parcimonieuses, car les supports définissent les frontières autour des exemples. Cet aspect se révèle crucial lorsque l'application envisagée nécessite une fonction de prédiction à taille limitée, afin de répondre en temps réel à un traitement. L'algorithme *Least Angle Regression Stepwise* [2] est une méthode à noyaux parcimonieuse, mais qui nécessite un paramètre pour régler le compromis biais-variance. Ce réglage peut s'avérer particulièrement délicat car le compromis peut être difficile à effectuer et nécessite un choix extérieur qui ne peut pas être automatisé dans certains cas.

Nous nous intéressons ici à la méthode *Relevance Vector Machine* introduite par Tipping en 2000 [9]. Cet algorithme probabiliste reprend certains aspects de l'approche du SVM, notamment par la forme de la fonction de décision. Cependant, alors que la résolution d'un SVM implique un paramètre ($C$) qui vise à régler la complexité de la solution, la résolution d'un RVM ne nécessite aucun paramètre autre que le choix du noyau. De plus, la solution obtenue en pratique est considérée comme parcimonieuse et présente des performances comparables à celle issue d'un SVM.

Nous proposons d'étendre la formulation des RVM aux noyaux multiples [4], afin de pouvoir définir une solution de la forme :

$$f(x) = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{i,j} K_j(x, x_i).$$

Cette formulation permet ainsi d'utiliser des noyaux différents pour chaque point de la solution, dans un but d'optimalité. Nous montrerons ainsi que la formulation RVM se prête naturellement à l'application de noyaux multiples.

Afin de valider cette approche, nous proposons d'utiliser cette approche pour des problèmes de sélection de variables, opérée ici en définissant un noyau spécifique par variable, et ce pour toutes les données. Ainsi, les solutions obtenues mettent en évidence l'adaptation spécifique pour chaque donnée support d'un ensemble de noyaux, c'est à dire d'un groupe de variables spécifique à chaque donnée support. Nous avons ainsi comparé deux approches différentes de noyaux multiples, en l'occurence l'algorithme SVM-MKL qui utilise un noyau composite et l'algorithme RVM-KB. Les résultats obtenus ont permis de démontrer la pertinence d'une approche noyau multiple de type *Kernel Basis* associée à l'algorithme RVM, qui obtient des performances équivalentes et parfois supérieures à celle du SVM-MKL, avec cependant un avantage non négligeable concernant la parcimonie de la solution en faveur de l'approche RVM.

## 2 Relevance Vector Machine

L'algorithme RVM est un modèle probabiliste parcimonieux qui considère un ensemble de $n$ données $\{\mathbf{x}_i, y_i\}_{i=1\ldots n}$, avec $\mathbf{x}$ un vecteur de dimension $d$, associé à l'étiquette $y$. Ce modèle a été défini initialement pour la régression en déterminant ainsi la probabilité $p(y|\mathbf{x}) \sim \mathcal{N}(f(\mathbf{x}), \sigma^2)$, où $\sigma$ correspond à la variance du bruit ajouté aux données. Le principe consiste à deviner la distribution de probabilité sousjacente qui génère les données : $p(\mathbf{y}|\mathbf{w}, \sigma^2) = \frac{1}{(\sigma\sqrt{2\pi})^n} \exp^{-\frac{1}{2\sigma^2}\|\mathbf{y}-\Phi\mathbf{w}\|^2}$, avec $\Phi$ une matrice comprenant le noyau.

La fonction de prédiction obtenue est de la forme :

$$f(\mathbf{x}) = \sum_{i=1}^{n} w_i \cdot \Phi(\mathbf{x}, \mathbf{x}_i) + w_0,$$

avec $\mathbf{w}$ les cœfficients associés à chaque donnée support.

Tipping s'appuie ici sur l'*Automatic Relevance Determination* proposée par Mackay[5], dont le principe revient à éliminer les paramètres qui ne permettent pas de définir la solution. Ainsi, pour chaque cœfficient $w$ de la fonction de décision, le prior associé est un hyperparamètre dont la valeur évolue durant la phase d'optimisation. En suivant le principe du rasoir d'Occam, les paramètres éliminés en priorité sont ceux qui ne contribuent pas à maximiser la densité de probabilité des paramètres selon l'ensemble des données. Ainsi, les paramètres qui ajoutent une complexité trop importante et ne permettent pas de maximiser de manière globale la densité de probabilité sont éliminés de la solution. L'ensemble de vecteurs qui permet finalement de construire la solution est alors suffisante pour définir l'ensemble des points.

La résolution a été détaillée longuement dans l'article original [9] et présente également l'adaptation de cet algorithme à la discrimination.

Le principal inconvénient de l'utilisation de noyaux est le réglage nécessaire lorsque les fonctions comportent des paramètres. Tipping propose ainsi d'appliquer une étape intermédiaire, de type *Expectation-Maximisation*, afin de régler les noyaux, mais le même paramètre est appliqué pour l'ensemble des données, ce qui n'est pas optimal car la solution peut être composée en réalité d'un noyau global associé à d'autres noyaux locaux. La finalité revient ainsi à associer pour chaque point support de la solution un ensemble de noyaux qui soit spécifique à ce support. Ainsi, l'idéal consiste à pouvoir dissocier pour chaque point support un ensemble de noyaux issus de fonctions différentes ou d'ensembles de variables distincts.

En 2004, Bach et al. [1] ont proposé de formuler le noyau du SVM comme un noyau composite, c'est à dire une combinaison linéaire de noyaux :

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{k} \beta_j \cdot K_j(\mathbf{x}, \mathbf{x}_i) + b.$$

avec $\alpha$ les cœfficients associés aux vecteurs supports et $\beta$ les cœfficients de pondérations des noyaux. Cette approche permet ainsi d'optimiser chaque noyau individuellement et d'obtenir un noyau unique applicable sur l'ensemble des données support. Dans le cadre probabiliste, une telle approche de noyau composite existe également ([3]).

Cependant, elle ne permet pas d'appliquer pour chaque donnée support un ensemble de noyaux spécifiques à cette donnée support, qui se traduirait par :

$$f(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{i,j} \cdot \Phi_j(\mathbf{x}, \mathbf{x}_i) + w_0.$$

Cette formulation de noyau multiple se ramène à une approche de type *Kernel Basis* [11], où le noyau $\Phi$ est décomposé en $k$ blocs de noyaux :

$$\Phi = \left[ \begin{array}{cccc} K_1 & K_2 & \ldots & K_k \end{array} \right].$$

La matrice $\Phi$ est ainsi de dimension $n \times (n \times k)$ et nous associons à chaque colonne de cette matrice un poids $w_i$ dans la fonction de décision : $f(\mathbf{x}) = \sum_{i=1}^{n*k} w_i \cdot \phi_i(\mathbf{x}) + w_0$. Comme nous le constatons, il est possible d'utiliser cette décomposition de noyaux au sein de l'algorithme RVM, qui déterminera ensuite la valeur des cœfficients $w_i$ associés à chaque colonne.

## 3 Résultats : application à la sélection de variables

Nous appliquons ici les noyaux multiples à une approche de sélection de variables, afin d'illustrer cette approche. Nous comparons les performances par rapport à l'algorithme SVM-MKL, avec l'implémentation proposée par Rakotomamonjy et al. [7] que nous avons eu l'occasion

d'utiliser précédemment dans un cadre de fusion de descripteurs. Les résultats sont obtenus sur 2 bases publiques du site UCI (`http://archive.ics.uci.edu/ml/`) fréquemment utilisées en apprentissage : AUTO MPG et BOSTON Housing.

La procédure est la suivante : pour chaque base, nous effectuons une validation croisée de 4 ensembles afin d'obtenir la meilleure performance pour un noyau simple. Ensuite, nous définissons le noyau multiple en calculant un noyau pour chaque variable. Ainsi, pour $n$ données de dimension $d$, l'algorithme SVM-MKL fusionnera $d$ noyaux de taille $n \times n$, c'est à dire optimisera $d$ cœfficients du noyau composite et $n$ cœfficients de pondération des points supports. L'algorithme RVM-KB prend en entrée une matrice de taille $n \times (n \times d)$ et calculera donc $(n \times d)$ cœfficients de pondération des points supports. La comparaison du nombre de points supports (la taille d'une solution de validation croisée) doit donc considérer le nombre de noyaux multiplié par le nombre de vecteur supports du SVM par rapport au nombre de points supports de la solution RVM.

L'erreur estimée est l'erreur quadratique sur l'ensemble des données. Dans le cas noyau multiple, nous utilisons les mêmes paramètres de noyau pour l'ensemble des variables, l'idéal étant de pouvoir associer un paramètre noyau spécifique pour chaque variable, mais nous sommes limités par la mémoire nécessaire pour ce type d'approche. Comme les variables sont normalisées, ce compromis est pertinent.

Un aspect peut également s'avérer déterminant dans le contexte de l'apprentissage afin d'évaluer le temps nécessaire pour apprendre un modèle. Ce facteur n'a pas été spécifiquement évalué ici car les deux approches ne sont pas comparables. L'algorithme SVM est très rapide comparativement à l'approche RVM, mais ce dernier présente l'avantage de ne pas nécessiter de réglage d'hyperparamètre qui handicape ainsi l'approche SVM. De façon qualitative, l'apprentissage reste cependant plus rapide par l'algorithme SVM-MKL. En contrepartie, le temps pour la prédiction est beaucoup plus court grâce à la parcimonie du RVM.

## 3.1 Base de données AUTO

Le premier jeu de données AUTO contient des informations sur la consommation de 398 véhicules avec 7 variables apportant des informations sur les caractéristiques du véhicule. Nous reportons dans le tableau 1 les conclusions du test complet où nous avons évalué les paramètres de noyau optimaux, ainsi que pour le SVM qui implique le paramètre $C$ et le paramètre $\epsilon$ qui gère la largeur du tube. Pour un seul noyau, l'approche SVM est plus performante, mais au prix d'un plus grand nombre de points supports. Avec les noyaux multiples, l'algorithme RVM-KB obtient la même erreur que SVM-MKL, mais ne retient que 101 points supports (pour un noyau initial de taille $298 \times 2388$), soit 101 scalaires, alors que la solution SVM implique 292 points et 6 noyaux.

| AUTO | Noyau | erreur | #RV (%) |
|---|---|---|---|
| RVM | Gaussien, 2 | 2.84 | 20 (6.7) |
| RVM-KB | Poly, 3 | 2.70 | 101 (4.2) |

| AUTO | Paramètres | Noyau | erreur | #SV (%) |
|---|---|---|---|---|
| SVM | $\epsilon = 0.2$<br>$C = 100$ | Gaussien, 3 | 2.74 | 272 (91) |
| SVM-MKL | $\epsilon = 0.1$<br>$C = 10$ | Gaussien, 1 | 2.70 | 292 (97) |

TABLE 1 – Performances obtenues sur la base AUTO, avec le nombre de points supports pour chaque solution, selon les paramètres de noyau optimaux.

| | | RVM-KB | | SVM-MKL |
|---|---|---|---|---|
| noyau | Variable(s) | #RV | $\sum_i \lvert w_i \rvert$ | $\beta$ |
| 1 | # cylindres | 18.25 | 0.1886 | 0.0081 |
| 2 | distance | 13.25 | 0.0596 | 0.0323 |
| 3 | puissance | 10.25 | 0.2772 | 0.0289 |
| 4 | poids | 7.75 | 0.2242 | 0.1053 |
| 5 | accélération | 1 | 0.0029 | 0 |
| 6 | année | 38.25 | 0.2147 | 0.0724 |
| 7 | origine | 0 | 0 | 0 |
| 8 | [1-7] | 12.25 | 0.0328 | 0.7530 |

TABLE 2 – Pondération associée à chaque variable selon les différentes stratégies RVM-KB ou SVM-MKL.

Nous détaillons ensuite dans le tableau 2 les variables retenues par les RVM-KB et SVM-MKL. Pour le SVM, nous affichons le poids $\beta$ associé à chaque noyau, dans le noyau composite final. Dans le cas du RVM, nous affichons le nombre de points supports retenus pour une variable donnée et dans la quatrième colonne le poids total associé à chaque variable dans la solution en cumulant les poids $w$ de chaque point support. Nous constatons que la méthode SVM accorde davantage d'importance au noyau 8 qui comprend l'ensemble des variables ($\beta = 0.75$), mais nous notons de nombreuses similarités pour les autres variables, en particulier les variables 2, 5 et 7 qui sont négligées dans les deux cas. En se rapportant à l'objectif d'estimation de la consommation d'un véhicule, ces variables semblent effectivement moins pertinentes que d'autres telles que le poids, la cylindrée ou la puissance.

## 3.2 Base Boston Housing

Le deuxième test porte sur le jeu Boston Housing, qui décrit le prix immobilier de 506 maisons en fonction de 13 variables. Ici, les résultats du tableau 3 montrent une légère amélioration pour les SVM et RVM en noyau multiple, avec un gain significatif pour les RVM-KB qui sont plus performants tout en restant plus parcimonieux.

| BOSTON | Noyau | erreur | #RV (%) |
|---|---|---|---|
| RVM | Gaussien,5 | 3.81 | 30 (7.8) |
| RVM-KB | Gaussien,2 | 3.22 | 322 (6.1) |

| BOSTON | Paramètres | Noyau | erreur | #SV (%) |
|---|---|---|---|---|
| SVM | $\epsilon = 0.05$<br>$C = 100$ | Gaussien,3 | 3.27 | 373 (98) |
| SVM-MKL | $\epsilon = 0.01$<br>$C = 10$ | Gaussien,1 | 3.26 | 375 (98) |

TABLE 3 – Performances obtenues sur la base BOSTON, avec le nombre de points supports pour chaque solution, selon les paramètres de noyau optimaux.

En considérant l'influence des variables dans le tableau 4, il faut noter que le noyau comprenant l'ensemble des variables est prédominant, en SVM et RVM, mais dans ce dernier, l'apport de noyaux dans la solution ne comprenant qu'une variable permet d'améliorer les performances. Les comportements des deux algorithmes sont relativement similaires car le RVM-KB et le SVM-MKL accordent moins d'importance aux variables 2, 3, 4, 7, 8 et 9. L'approche noyau composite est ici capable de supprimer complètement un noyau ($\beta = 0$), alors que l'approche *Kernel Basis* permet de conserver quelques représentants locaux de ces variables, mais avec une importance vraiment faible.

Dans cet exemple également la parcimonie des RVM permet de contenir la taille de la solution, avec seulement 322 points contre 375*7 pour les SVM-MKL.

| | | RVM-KB | | SVM-MKL |
|---|---|---|---|---|
| Noyau | Variable(s) | #RV | $\sum_i \lvert w_i \rvert$ | $\beta$ |
| 1 | Taux criminalité | 5.75 | 0.0050 | 0.0040 |
| 2 | % residences | 74.25 | 0.0005 | 0 |
| 3 | % industries | 13 | 0.0014 | 0 |
| 4 | Rivière | 11.75 | 0.0006 | 0 |
| 5 | % NOx | 16 | 0.0037 | 0.0104 |
| 6 | # chambres | 15 | 0.0035 | 0.1271 |
| 7 | Age | 3.5 | 0.0008 | 0 |
| 8 | Distance | 0 | 0 | 0 |
| 9 | Accès autoroute | 61 | 0.0200 | 0 |
| 10 | Taxe foncière | 21.25 | 0.0027 | 0.0181 |
| 11 | Ecoles | 10.7500 | 0.0016 | 0.0185 |
| 12 | % immigrés | 1.2500 | 0.0003 | 0 |
| 13 | Niveau social | 14.7500 | 0.0085 | 0.0939 |
| 14 | [1-13] | 73.7500 | 0.9513 | 0.7281 |

TABLE 4 – Pondération associée à chaque variable selon les différentes stratégies RVM-KB ou SVM-MKL.

# 4 Conclusion

Nous avons présenté dans ce papier une extension des RVM aux noyaux multiples de type *Kernel Basis*. Comparativement à l'approche de noyaux multiples de type noyau composite, le *Kernel Basis* permet de combiner des noyaux globaux et locaux, c'est à dire d'appliquer spécifiquement à chaque vecteur de la solution un ensemble de noyaux spécifique.

Cette approche est particulièrement utile dans des tâches de sélection de modèle et nous avons illustré cet intérêt sur des exemples de sélection de variable. Les performances obtenues sont comparables, voire meilleures, par rapport à un SVM utilisant un noyau composite, avec cependant un avantage important en faveur des RVM qui se révèlent beaucoup plus parcimonieux en pratique.

La limitation des RVM se situe cependant dans la complexité lors de l'apprentissage qui nécessite une mémoire importante et ne permet donc pas de constituer un ensemble initial contenant beaucoup de noyaux. Cet aspect sera ainsi une perspective d'amélioration afin d'exploiter au mieux cette approche.

# Références


[1] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *ICML'04 : Proceedings of the twenty-first international conference on Machine learning*, page 6, New York, NY, USA, 2004. ACM Press.

[2] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. pages 407–499, january 2003.

[3] M. Girolami and S. Rogers. Hierarchic bayesian models for kernel learning. In $22^{nd}$ *International Conference on Machine Learning*, pages 241–248, 2005.

[4] S. Gunn and J. Kandola. Structural modelling with sparse kernels. In *Machine Learning*, volume 48, pages 137–163, 2002.

[5] D. J. Mackay. Probable networks and plausible predictions - a review of praticta bayesian methods for supervised neural networks. 6 :469–505, 1995.

[6] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. J. Smola, and K.-R. Mueller. Constructing descriptive and discriminative non-linear features : Rayleigh coefficients in kernel feature spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[7] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. Simple MKL. In *Journal of Machine Learning Research*, 2008.

[8] B. Schölkopf and A. J. Smola. *Learning with Kernels.* MIT Press, 2002.

[9] M. Tipping. The relevance vector machine. In T. K. L. S. A. Solla and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000.

[10] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, N.Y, 1995.

[11] P. Vincent and Y. Bengio. Kernel matching pursuit. *Mach. Learn.*, 48(1-3) :165–187, 2002.