Approximation de matrices pour l'apprentissage des hyperparamètres des fonctions noyaux Gaussiennes

Mathieu FAUVEL[1]

[1]University of Toulouse, ENSAT, UMR 1201 DYNAFOR, INRA & , F-31326 Castanet Tolosan, France
mathieu.fauvel@ensat.fr

Résumé – Le problème considéré dans cet article concerne l'optimisation des hyperparamètres d'une fonction noyau Gaussienne à l'aide de mesures de similitude entre matrices. Deux contributions sont proposées : 1) une nouvelle mesure de similarité entre fonctions noyaux et 2) une nouvelle paramétrisation pour les noyaux Gaussiens. Des améliorations des temps de calculs et des taux de bonnes classifications par rapport à la validation croisée pour un classifier k-nn sont obtenues sur des jeux de données standards.

Abstract – The problem considered in this paper concerns the optimization of the hyperparameters of a Gaussian kernel function using a similarity measure between matrices. Two contributions are proposed: 1) a new measure of similarity between kernel functions and 2) a new parameterization for the Gaussian kernel. Improvements in terms of computation time and classification accuracies by comparison to cross-validation are obtained on standard data set for a k-nn classifier.

1 Introduction

Les méthodes à noyaux ont montré des performances remarquables dans de nombreux domaines d'applications [1]. Si un large choix d'algorithmes existe pour la classification, la régression, l'estimation de problèmes inverses ou pour le démélange spectral [2, 3], la qualité des résultats dépend grandement de la fonction noyau utilisée ainsi que du réglage des hyperpapramètres du noyau. En pratique, pour une grande majorité de jeux de données, le noyau radial donne de très bon résultats, sinon les meilleurs. Il s'écrit :

$$k(\mathbf{x},\mathbf{z}) = \exp\left(-\gamma d\left(\mathbf{x},\mathbf{z}\right)\right) \tag{1}$$

où d est une semi-métrique entre $\mathbf{x}$ et $\mathbf{z}$. Toutes les semi-métriques peuvent être utilisées en fonction des propriétés des données [4], mais on a très souvent $\mathbf{x} \in \mathbb{R}^d$, $\gamma > 0$, $d(\mathbf{x},\mathbf{z}) = \|\mathbf{x}-\mathbf{z}\|^2$ et il suffit de rechercher la valeur optimale de l'hyperparamètre d'échelle γ. Pour cela, des techniques comme la validation croisée [5] permettent une bonne estimation de la valeur optimale de γ. Cependant, la validation croisée peut être difficile à mettre en œuvre en pratique : temps de calculs importants, définition de l'intervalle de recherche, faible nombre d'hyperparamètres ... Or, il a été montré que l'utilisation de métriques du type $d\left(\mathbf{x},\mathbf{z}\right) = (\mathbf{x}-\mathbf{z})^t\mathbf{Q}(\mathbf{x}-\mathbf{z})$ donnait des résultats intéressants par rapport à une métrique Euclidienne [6, 7]. Plusieurs stratégies ont été proposées pour l'optimisation de la matrice $\mathbf{Q}$ et deux types d'approches peuvent être identifiés :

- Minimisation d'une estimation régularisée de l'erreur, nécessitant d'optimiser l'algorithme à noyau (classification SVM, régression ridge ...) [6, 8, 9],
- Minimisation d'une erreur d'approximation entre le noyau utilisé et un noyau idéal pour un traitement donné (alignement de matrices ...) [10].

L'avantage de la seconde approche vient du fait qu'il n'est pas nécessaire d'apprendre la règle de décision lors de l'optimisation des hyperparamètres, ce qui réduit la charge calculatoire et permet de ne pas restreindre $\mathbf{Q}$ à un scalaire ou à une matrice diagonale.

Dans cet article, nous proposons d'une part une nouvelle mesure pour l'approximation de matrice noyau, basée sur la distance de Frobenius entre matrices, et d'autre part une paramétrisation de $\mathbf{Q}$, basée sur la factorisation de Choleski, permettant de définir une métrique. Dans la suite de l'article, nous nous limiterons à la classification, mais il est possible d'étendre les concepts présentés à d'autres problèmes d'apprentissage comme la régression.

2 Approximation de matrice noyau

L'approximation de noyau à l'aide de l'alignement a été proposée par [10]. L'idée est d'approcher une matrice noyau idéale à l'aide d'une famille de noyaux paramétrées par les hyperparamètres à optimiser (par exemple, l'hyperparamètre d'échelle pour le noyau Gaussien). On notera $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ l'ensemble des échantillons d'entraînement, où y_i représente le label de la classe.

Le noyau Gaussien (1) est une mesure de similitude entre deux échantillons $\mathbf{x}_i$ et $\mathbf{x}_j$: elle vaut 1 dans le cas où les deux échantillons sont très proches ou identiques, dans le sens de la distance utilisée, et elle vaut 0 lorsque les échantillons sont très éloignés. On définit le comportement de la fonction noyau *idéale* k_I comme : $k_I(\mathbf{x}_i,\mathbf{x}_j) = 1$ si $y_i = y_j$ et 0 sinon.

2.1 Mesure de similitude entre matrices

L'alignement est couramment utilisé pour mesurer la similitude entre deux noyaux [11]. En notant $\mathbf{K}$ la matrice de Gram des évaluations de la fonction noyau k, paramétrée par $\boldsymbol{\sigma}$, l'alignement

(empirique) s'écrit :

$$\mathcal{A}(\boldsymbol{\sigma}, \mathcal{S}) = \frac{\langle \mathbf{K}, \mathbf{K}_I \rangle_F}{\|\mathbf{K}\|_F \|\mathbf{K}_I\|_F}. \quad (2)$$

$\langle \cdot, \cdot \rangle_F$ représente le produit scalaire de Frobenius et $\|\cdot\|_F$ la norme associée. Dans [12], nous avons proposé la distance (empirique) induite par le produit scalaire de Frobenius comme mesure de similitude :

$$\mathcal{F}(\boldsymbol{\sigma}, \mathcal{S}) \quad = \quad \frac{\|\mathbf{K} - \mathbf{K}_I\|_F^2}{n^2}. \quad (3)$$

$\mathcal{A}$ est compris entre -1 et 1 et $\mathcal{F}$ est compris entre 0 et 1. Pour l'optimisation, on cherche à maximiser $\mathcal{A}$ et à minimiser $\mathcal{F}$. Notons que (2) est une semi-métrique tandis que (3) est une métrique. Pour le problème d'optimisation, cela implique la résolution d'un problème non-convexe dans le premier cas et convexe dans le second cas. Par ailleurs, (3) possède le même propriété que (2) pour la concentration [11] :

Théorème 1. *Pour une fonction noyau Gaussienne, l'estimateur empirique de la distance $\mathcal{F}$ est concentré autour de sa valeur moyenne.*

$$p\left\{\mathcal{S} : \left|\mathcal{F}(\boldsymbol{\sigma}, \mathcal{S}) - \mathcal{F}(\boldsymbol{\sigma})\right| \geq \hat{\epsilon}\right\} \leq \delta. \quad (4)$$

Démonstration. On utilise l'inégalité de McDiarmid pour démontrer (4). On note $\mathcal{F}(\boldsymbol{\sigma}) = \mathbb{E}_{\mathcal{S}}\left[\mathcal{F}(\boldsymbol{\sigma}, \mathcal{S})\right]$ et $\mathcal{S}' = \mathcal{S} \backslash (\mathbf{x}_t, y_t) \cup (\mathbf{x}'_t, y'_t)$. On peut vérifier que

$$\left|\mathcal{F}(\boldsymbol{\sigma}, \mathcal{S}) - \mathcal{F}(\boldsymbol{\sigma}, \mathcal{S}')\right| \quad \leq \quad \frac{2}{n}$$

pour tout $t \in \{1, \ldots, n\}$. D'après l'inégalité de McDiarmid, on obtient

$$p\left\{\mathcal{S} : \left|\mathcal{F}(\boldsymbol{\sigma}, \mathcal{S}) - \mathcal{F}(\boldsymbol{\sigma})\right| \geq \epsilon\right\} \leq 2\exp\left(-\frac{n\epsilon^2}{2}\right) \quad (5)$$

On obtient (4) en posant $\epsilon = \hat{\epsilon} = \sqrt{\frac{2(\ln(2) - \ln(\delta))}{n}}$. □

Finalement, (2) et (3) sont optimisées à l'aide d'une méthode de Newton classique [13]. Le gradient de (2) et de (3) doit être calculé lors de l'optimisation. Le produit de Frobenius étant un opérateur linaire, les fonctions dérivées s'écrivent simplement en fonction de la dérivée du noyau :

$$\frac{\partial \mathcal{A}(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} \quad = \quad \frac{1}{\|\mathbf{K}_I\|_F}\left[\frac{\left\langle \mathbf{K}_I, \frac{\partial \mathbf{K}}{\partial \boldsymbol{\sigma}} \right\rangle_F}{\|\mathbf{K}\|_F} - \frac{\langle \mathbf{K}, \mathbf{K}_I \rangle_F \left\langle \mathbf{K}, \frac{\partial \mathbf{K}}{\partial \boldsymbol{\sigma}} \right\rangle_F}{\|\mathbf{K}\|_F^{3/2}}\right] \quad (6)$$

$$\frac{\partial \mathcal{F}(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} \quad = \quad \frac{2}{n^2}\left\langle \mathbf{K} - \mathbf{K}_I, \frac{\partial \mathbf{K}}{\partial \boldsymbol{\sigma}} \right\rangle_F. \quad (7)$$

2.2 Noyau Gaussien

Le noyau Gaussien paramété par $\mathbf{Q}$ s'écrit

$$k(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{z})^t \mathbf{Q}(\mathbf{x} - \mathbf{z})\right) \quad (8)$$

où $\mathbf{Q}$ est une matrice carrée de taille d définie positive. Nous proposons trois paramétrisations pour $\mathbf{Q}$, les deux premières étant classiques dans la littérature. La troisième se distingue des paramétrisations proposées par le fait que la matrice est définie positive, et non semi-définie positive [14]. Les trois paramétrisations sont dénommées *sphérique*, *elliptique* et de *Choleski* :

1. $\mathbf{Q} = \sigma^2 \mathbf{I}$ avec $\sigma > 0$. Pour simplifier le problème d'optimisation en éliminant la contrainte de positivité, nous proposons d'utiliser la paramétrisation suivante : $\mathbf{Q} = \exp(\sigma)\mathbf{I}$. Sous cette paramétrisation, la dérivée de la fonction noyau s'écrit :

$$\begin{aligned} \frac{\partial k(\mathbf{x}, \mathbf{z})}{\partial \sigma} &= -\frac{1}{2}\exp(\sigma)\|\mathbf{x} - \mathbf{z}\|^2 k(\mathbf{x}, \mathbf{z}) \\ &= \log\big(k(\mathbf{x}, \mathbf{z})\big) k(\mathbf{x}, \mathbf{z}) \end{aligned} \quad (9)$$

2. $\mathbf{Q} = \mathrm{diag}[\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2]$. Là encore, pour éliminer les contraintes de positivité, la paramétrisation suivante est utilisée :

$$\mathbf{Q} \quad = \quad \mathrm{diag}[\exp(\sigma_1), \exp(\sigma_2), \ldots, \exp(\sigma_d)].$$

On obtient la dérivée de la fonction noyau par :

$$\frac{\partial k(\mathbf{x}, \mathbf{z})}{\partial \sigma_p} \quad = \quad -\frac{\big(\mathbf{x}[p] - \mathbf{z}[p]\big)^2}{2}\exp(\sigma_p) k(\mathbf{x}, \mathbf{z}) \quad (10)$$

3. $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$ avec $\mathbf{L}$ une matrice triangulaire inférieure dont les éléments diagonaux sont strictement positifs. $\mathbf{L}$ est la factorisation de Choleski de $\mathbf{Q}$. En imposant la positivité pour les éléments diagonaux, cette décomposition est unique et $\mathbf{Q}$ est définie positive.

$$\mathbf{L}^T = \begin{pmatrix} \exp(\sigma_{11}) & \sigma_{21} & \ldots & \sigma_{d1} \\ 0 & \exp(\sigma_{22}) & \ldots & \sigma_{d2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \exp(\sigma_{dd}) \end{pmatrix} \quad (11)$$

La dérivée de la fonction noyau s'obtient de la manière suivante ($1 \leq m \leq l \leq d$ et $\delta_{lm} = 1$ si $l = m$ et 0 sinon) :

$$\begin{aligned} &\frac{\partial k(\mathbf{x}, \mathbf{z})}{\partial \sigma_{lm}} = \frac{\partial}{\partial \sigma_{lm}} \exp\Big(-\frac{1}{2}\|\mathbf{L}^T(\mathbf{x} - \mathbf{z})\|^2\Big) \\ &= -\frac{1}{2}\frac{\partial \|\mathbf{L}^T(\mathbf{x} - \mathbf{z})\|^2}{\partial \sigma_{lm}} k(\mathbf{x}, \mathbf{z}) \\ &= -\Big[\exp(\sigma_{lm})(\mathbf{x}[l] - \mathbf{z}[l])\delta_{lm} + (\mathbf{x}[m] - \mathbf{z}[m])(1 - \delta_{lm})\Big] \\ &\quad \times \Big[\exp(\sigma_{ll})(\mathbf{x}[l] - \mathbf{z}[l]) + \sum_{i=l+1}^{d} \sigma_{li}(\mathbf{x}[i] - \mathbf{z}[i])\Big] k(\mathbf{x}, \mathbf{z}). \end{aligned} \quad (12)$$

3 Expériences

3.1 Données UCI

Des expériences ont été menées sur des données de l'UCI Machine Learning Repository (`http://archive.ics.uci-.edu/ml/`). Un k-nn à noyau a été utilisé comme classifieur. Un classifieur plus évolué (type SVM) ne permettrait pas de voir l'influence des différentes paramétrisations sur les résultats de classification. Pour chaque jeu de données, deux tiers des échantillons ont été utilisés pour l'entraînement et le tiers restant a été utilisé pour la validation. 50 répétitions ont été effectuées pour chaque jeu de données, la moyenne et l'écart type de la précision globale

TABLE 1 – Résultats de classification en termes de pourcentage d'échantillons bien classés. n représente le nombre d'échantillons et d le nombre de variables. s, e, et c représentent respectivement les paramétrisations sphérique, elliptique et de Choleski. Les valeurs entre parenthèses correspondent à l'écart type sur les 50 répétitions (20 pour flc1). Pour $\mathcal{A}$ et $\mathcal{F}$ est retournée la valeur de la mesure de similitude en fonction de la paramétrisation.

Données			**Alignement**				**Distance de Frobenius**				**vc**
	n	d	s	e	c	$\mathcal{A}$	s	e	c	$\mathcal{F}$	s
Iris	150	4	94.9 (2.7)	95.8 (2.8)	**96.0 (2.1)**	0.84, 0.90, 0.93	94.9 (2.7)	95.8 (2.9)	96.0 (2.3)	0.10, 0.06, 0.04	93.3 (3.7)
Wine	178	13	94.2 (2.6)	96.5 (2.0)	**97.6 (2.2)**	0.80, 0.87, 0.94	94.2 (2.6)	95.9 (2.0)	97.5 (2.3)	0.13, 0.09, 0.04	92.9 (4.6)
Ionosphere	351	34	86.3 (2.7)	90.6 (2.4)	89.2 (2.4)	0.76, 0.81, 0.88	86.3 (2.7)	**90.9 (2.3)**	89.3 (2.6)	0.23, 0.18, 0.13	83.9 (8.0)
Diabete	768	8	**70.1 (1.9)**	68.9 (2.4)	68.9 (2.6)	0.75, 0.75, 0.76	**70.1 (1.9)**	69.9 (2.4)	69.7 (2.5)	0.25, 0.24, 0.23	68.5 (5.6)
Cancer	683	10	**95.7 (1.1)**	95.3 (1.4)	95.3 (1.1)	0.91, 0.92, 0.95	**95.7 (1.1)**	95.1 (1.4)	95.2 (1.2)	0.10, 0.09, 0.06	95.1 (3.4)
Flc1	70810	9	92.2 (0.2)	92.4 (0.3)	92.4 (0.2)	0.67, 0.69, 0.69	92.2 (0.2)	**92.5 (0.3)**	92.2 (0.4)	0.06, 0.06, 0.06	91.9 (0.6)

sont reportés. Pour la paramétrisation sphérique, l'approche proposée pour la sélection de l'hyperparamètre a été comparée à la validation croisée. Les résultats sont donnés dans la Table 1.

Excepté pour *Diabete*, les paramétrisations elliptique et Choleski permettent une amélioration des résultats en termes de précision globale, avec des performances similaires pour $\mathcal{A}$ et $\mathcal{F}$. Pour *Diabete*, $\mathcal{F}$ donne de meilleurs résultats que $\mathcal{A}$ pour les deux paramétrisations elliptique et Choleski. Pour ce jeu de données, on peut voir que les valeurs de similitude sont très proches quelle que soit la paramétrisation utilisée. Dans ce cas là, autres paramétrisation n'apportent pas d'amélioration par rapport à la paramétrisation sphérique. Pour ces jeux de données et le classifieur k-nn, la validation croisée donne les moins bons résultats. On remarque une bonne corrélation entre les valeurs hautes (basses) de $\mathcal{A}$ ($\mathcal{F}$) et le taux de bonnes classifications.

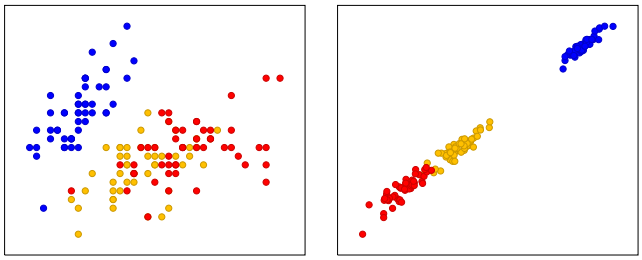
Les résultats obtenus sur les 50 répétitions ont été soumis au test de Wilcoxon pour valider statistiquement les différences observées. Pour le jeu de données *Iris*, les différences entre les résultats obtenus pour les différentes paramétrisations ne sont pas significatives. Par contre, elles le sont toutes par rapport à la validation croisée. Pour les autres jeux de données, les résultats sont reportés dans la Table 2. Globalement, on peut remarquer que pour ces 4 jeux de données les différences entre les résultats obtenus pour les critères $\mathcal{A}$ et $\mathcal{F}$ pour un même modèle (s, e et c) ne sont pas significatifs, l'amélioration est plutôt dans les temps de calculs (voir paragraphe suivant). Néanmoins, les résultats obtenus par les modèles proposés sont significativement meilleurs que ceux obtenus avec la validation croisée.

Le temps de calculs pour l'optimisation des modèles pour les données *Cancer* sur un PC portable standard sont pour les trois paramétrisations 0.14s, 0.38s et 5.90s pour $\mathcal{F}$ et 0.20s, 0.65s et 6.87s pour $\mathcal{A}$. L'amélioration des temps de calculs vient d'une part que le problème d'optimisation est convexe pour $\mathcal{F}$ et d'autre part que (3) et (7) sont plus simples à calculer que (2) et (6).

La Figure 1 illustre l'effet de la paramétrisation de Choleski sur les données Iris. Après transformation, les données sont davantage séparables linéairement.

3.2 Données satellitaires

Le jeu de données considéré ici est l'image multispectrale *Southern Tippecanoe County*, Indiana, USA, acquis par le capteur Flightline C1. Neuf classes ont été définies pour un total de 70810 pixels labélisés. 400 pixels par classe ont été extraits pour l'entraînement, le reste des pixels a été utilisé pour la validation. 20 répétitions ont été effectuées en tirant aléatoirement les pixels d'entrainements à chaque fois.



(a) (b)

FIGURE 1 – Scatter plot des deux premières variables pour les données Iris originales (a) et après projection sur les axes issus du modèle de Choleski (b). Chaque couleur représente une classe différente.

Les résultats en termes de pourcentage de bonnes classifications sont reportés dans la Table 1 et les tests de signification statistique dans la Table 2. Excepté pour $\mathcal{F}_c$, tout les modèles donnent des résultats légèrement supérieurs à ceux obtenus par validation croisée.

Les temps de calculs des paramètres sont donnés à la figure 2. Excepté pour $\mathcal{A}_s$ et $\mathcal{F}_s$, les différences sont toutes statistiquement significatives. Pour les modèles sphériques, les temps de calculs sont réduit d'un facteur dix par rapport à la validation croisée. Pour les modèles elliptique et de Choleski, optimiser $\mathcal{F}$ prend en moyenne deux fois moins de temps que d'optimiser $\mathcal{A}$. De plus, il est possible d'optimiser $\mathcal{F}_e$ pour un coût calculatoire moindre que la validation croisée.

4 Conclusion

Dans cet article, deux contributions ont été proposées pour l'optimisation des hyperparamètres d'une fonction noyau Gausienne. La première contribution est la définition d'une mesure de similarité basée sur la distance de Frobenius. La concentration de l'estimateur empirique de la distance de Frobenius a été étudié. Comparée à l'alignement de noyaux, la distance de Frobenius permet de mettre en œuvre une procédure d'optimisation convexe ce qui améliore les temps de calculs. La seconde contribution concerne la définition d'une paramétrisation basée sur la factorisation de Choleski. Combinée au critère de similitude, elle permet la défini-

TABLE 2 – Test de Wilcoxon pour les jeux de données UCI wine, Ionosphere, Diabete et Cancer, et les données flc1. Les cases blanches indiquent que les différences entre les précisions observées sur les 50 répétitions ne sont pas statistiquement significatives tandis que les cases grises indiquent que les différences sont significatives. Comme les tests sont symétriques, seule la partie supérieure/inférieure a été représentée : $a \backslash b$ signifie que les résultats pour le jeu de données a, respectivement b, sont les éléments non-diagonaux inférieurs, respectivement supérieur, du tableau.

	$\mathcal{A}_s$	$\mathcal{A}_e$	$\mathcal{A}_c$	$\mathcal{F}_s$	$\mathcal{F}_e$	$\mathcal{F}_c$	vc
$\mathcal{A}_s$							
$\mathcal{A}_e$							
$\mathcal{A}_c$							
$\mathcal{F}_s$							
$\mathcal{F}_e$							
$\mathcal{F}_c$							
vc							

(a) Ionosphere\wine

	$\mathcal{A}_s$	$\mathcal{A}_e$	$\mathcal{A}_c$	$\mathcal{F}_s$	$\mathcal{F}_e$	$\mathcal{F}_c$	vc
$\mathcal{A}_s$							
$\mathcal{A}_e$							
$\mathcal{A}_c$							
$\mathcal{F}_s$							
$\mathcal{F}_e$							
$\mathcal{F}_c$							
vc							

(b) Cancer\Diabete

	$\mathcal{A}_s$	$\mathcal{A}_e$	$\mathcal{A}_c$	$\mathcal{F}_s$	$\mathcal{F}_e$	$\mathcal{F}_c$	vc
$\mathcal{A}_s$							
$\mathcal{A}_e$							
$\mathcal{A}_c$							
$\mathcal{F}_s$							
$\mathcal{F}_e$							
$\mathcal{F}_c$							
vc							

(c) ~\ flc1

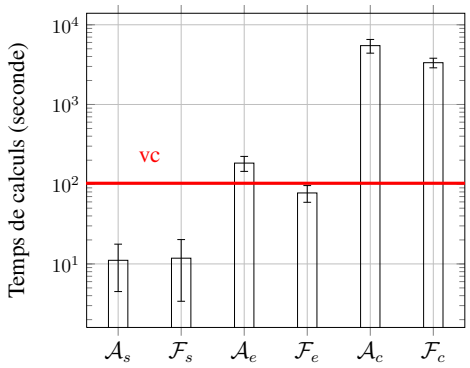


FIGURE 2 – Temps de calculs moyen et écart type pour les différent modèles pour les données satellitaires.

tion d'une métrique maximisant la séparation entre les différentes classes. En termes de précision de classification, les résultats obtenus sont significativement meilleurs que ceux obtenus par validation croisée.

Références

[1] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY, USA : Cambridge University Press, 2004.

[2] G. Camps-Valls and L. Bruzzone, eds., *Kernel Methods for Remote Sensing Data Analysis*. Wiley, 2009.

[3] J. Chen, C. Richard, and P. Honeine, "Nonlinear unmixing of hyperspectral data based on a linear-mixture/nonlinear-fluctuation model," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 480–492, 2013.

[4] B. Haasdonk and C. Bahlmann, "Learning with distance substitution kernels," in *Pattern Recognition* (C. Rasmussen, H. Bülthoff, B. Schölkopf, and M. Giese, eds.), vol. 3175 of *Lecture Notes in Computer Science*, pp. 220–227, Springer Berlin Heidelberg, 2004.

[5] T. Hastie, R. Tibshirani, and J. Friedman, eds., *The Elements of Statistical Learning, Data Mining, Inference, and Prediction*. Springer, 2009.

[6] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, pp. 131–159, 2002.

[7] G. Baofeng, S. Gunn, R. Damper, and J. Nelson, "Customizing kernel functions for SVM-based hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 17, pp. 622–629, Apr. 2008.

[8] Y. Grandvalet and S. Canu, "Adaptive scaling for feature selection in SVMs," in *NIPS*, pp. 553–560, 2002.

[9] H. Laanaya, F. Abdallah, H. Snoussi, and C. Richard, "Learning general gaussian kernel hyperparameters of SVMs using optimization on symmetric positive-definite matrices manifold," *Pattern Recognition Letters*, vol. 32, no. 13, pp. 1511 – 1515, 2011.

[10] C. Igel, T. Glasmachers, B. Mersch, N. Pfeifer, and P. Meinicke, "Gradient-based optimization of kernel-target alignment for sequence kernels applied to bacterial gene start detection," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 4, no. 2, pp. 216 –226, 2007.

[11] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor, "On kernel-target alignment," in *Advances in Neural Information Processing Systems 14*, pp. 367–373, MIT Press, 2002.

[12] M. Fauvel, "Kernel matrix approximation for learning the kernel hyperparameters," in *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pp. 5418 – 5421, july 2012.

[13] J. Bonnans, J. Gilbert, C. Lemaréchal, and C. Sagastizábal, *Numerical Optimization : Theoretical and Practical Aspects*. Springer, 2006.

[14] J.-B. Pothin and C. Richard, "Optimal feature representation for kernel machines using kernel-target alignment criterion," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 3, pp. III–1065 –III–1068, april 2007.