# Apprentissage de métrique appliqué à la classification de textes par méthodes à noyaux

Jean-Baptiste Pothin, Cédric Richard

Institut Charles Delaunay (ICD-M2S, FRE CNRS 2848)
Université de Technologie de Troyes, 12 rue Marie Curie, BP 2060, 10010 Troyes cedex - France
jean_baptiste.pothin@utt.fr, cedric.richard@utt.fr

**Résumé** – Dans cet article, nous proposons une méthode pour l'optimisation de la métrique d'un classifieur textuel à noyaux. Contrairement aux techniques populaires existantes, notre approche ne nécessite pas la définition explicite de règles sémantiques. Etant donné un ensemble d'apprentissage, l'algorithme proposé permet d'optimiser la matrice sémantique, sans qu'il soit nécessaire d'exhiber celle-ci. Les résultats expérimentaux montrent l'efficacité et l'utilité de la méthode proposée en classification de textes par SVM (Support Vector Machines).

**Abstract** – In this paper, we propose a method for optimizing the metric of a text-classifier, in a kernel setting. Unlike the most popular methods, our approach does not require the explicit design of semantic rules. Given a training set, the proposed algorithm learns the optimal semantic matrix, without requiring to exhibit it. Experimental results show that our method is effective and useful in text-classification by SVM (Support Vector Machines).

## 1 Introduction

Avec l'accroissement considérable du nombre de documents numériques, tels que les pages webs ou les courriels, les classifieurs textuels sont devenus des outils indispensables au traitement automatique de données. Les méthodes de classification de textes les plus répandues considèrent des documents représentés sous la forme de *sac de mots* [1]. Ces représentations, simples à mettre en œuvre, ne prennent en compte ni l'ordre des mots, ni leurs sémantiques. L'approche courante pour traiter ces défauts consiste à transformer les représentations à l'aide d'une *matrice sémantique*. Celle-ci est le plus souvent diagonale [2]. Le cas d'une matrice pleine est en général limité au concept de co-occurences des termes du corpus [3], et nécessite en pratique une réduction de la dimensionalité par ACP (Analyse en Composante Principale) [4].

Dans cet article, nous considérons la matrice sémantique comme un paramètre à *apprendre*. L'idée clé de notre approche repose sur le fait que la métrique d'un classifieur textuel est étroitement liée à sa matrice sémantique. L'optimisation de la métrique sera réalisé par l'algorithme $\mathrm{DA}_{\mathrm{lign}}$ [5], qui formule l'apprentissage sous la forme d'un problème d'optimisation convexe, similaire à celui des Support Vector Machines [6].

### 1.1 Noyaux *sac de mots*

Le principe de base pour classer des données de type texte consiste à représenter chaque document sous la forme d'un vecteur avant d'appliquer, dans l'espace généré, un algorithme pour données vectoriels [7]. La transformation la plus usitée, appelée communément *sac de mots*, représente un texte $x$ sous la forme suivante :

$$\phi : x \longmapsto \phi(x) = \big(\mathrm{tf}(t_1, x), \mathrm{tf}(t_2, x), \cdots, \mathrm{tf}(t_P, x)\big)^T \in \mathbb{R}^P,$$

où $\mathrm{tf}(t_i, x)$ est la fréquence d'occurrence du mot (ou *terme*) $t_i$ dans le document $x$. Ici, $\{t_i\}_1^P$ est un ensemble de $P$ termes appelé *dictionnaire*. Celui-ci est construit en relevant tous les termes apparaissant dans un ensemble de documents donnés appelé *corpus*.

Grâce à la transformation *sac de mots*, on détermine aisément un produit scalaire $\kappa$ entre deux documents $x_i$,$x_j$ :

$$\kappa(x_i, x_j) \triangleq \langle \phi(x_i), \phi(x_j) \rangle = \sum_{p=1}^{P} \mathrm{tf}(t_p, x_i)\mathrm{tf}(t_p, x_j). \quad (1)$$

Le dictionnaire comprenant typiquement plusieurs dizaines de milliers de mots, le calcul direct de (1) est en général très coûteux. En remarquant que la plupart des entrées de $\phi(x_1)$ et $\phi(x_2)$ sont nulles, il est néanmoins possible de réduire considérablement le temps de calcul. Il suffit pour cela de convertir chaque document en une liste comprenant les termes qu'il contient, puis de calculer :

$$\kappa(x_i, x_j) = A(L(x_i), L(x_j)), \quad (2)$$

où l'algorithme $A(\cdot, \cdot)$ traverse les listes $L(x_i)$ et $L(x_j)$ en calculant le produit des fréquences des termes communs. Cette procédure permet de déterminer la valeur de $\kappa(x_i, x_j)$ sans avoir à calculer explicitement les vecteurs $\phi$.

Par la suite, nous considérerons le cas d'un noyau $\kappa$ de norme unité. (1) n'étant clairement pas normalisé, il faudra appliquer la transformation suivante :

$$\kappa(x_i, x_j) \leftarrow \frac{\kappa(x_i, x_j)}{\sqrt{\kappa(x_i, x_i)\kappa(x_j, x_j)}}. \quad (3)$$

### 1.2 Noyaux sémantiques

Afin d'adapter le noyau *sac de mots* à la sémantique du corpus, l'approche courante consiste à choisir une matrice sémantique $\boldsymbol{S}$, de dimension $P \times P$, puis à appliquer la transformation

$$\tilde{\phi} : \phi(x) \mapsto \tilde{\phi}(x) = \boldsymbol{S}^T \phi(x). \tag{4}$$

Le produit scalaire s'exprime alors :

$$\begin{aligned} \langle \boldsymbol{S}^T \phi(x_i), \boldsymbol{S}^T \phi(x_j) \rangle &= \phi(x_i)^T \boldsymbol{S}\boldsymbol{S}^T \phi(x_j) & (5) \\ &= \tilde{\kappa}(x_i, x_j). & (6) \end{aligned}$$

Différentes matrices sémantiques ont été proposées dans la littérature [2, 3, 4]. La plus connue est la matrice diagonale appelée TF-IFD, de terme général :

$$w_p = w(t_p) = \ln\left(\frac{m}{\mathrm{df}(t_p)}\right), \tag{7}$$

où $m$ est le nombre de documents du corpus et $\mathrm{df}(t_p)$ le nombre de documents du corpus contenant le terme $t_p$. On constate que le poids $w_p$ est inversement proportionel à la fréquence d'occurence $\mathrm{df}(t_p)$. Dans le cas extrême où le terme $t_p$ est commun à tous les documents du corpus ($\mathrm{df}(t_p) = m$), le poids attribué est nul. Cette propriété est désirable dans le sens où $t_p$ apporte peu d'information discriminante dans ce cas. La fonction logarithme permet quant à elle ne pas attribuer trop d'importance aux termes n'apparaissant dans le corpus qu'occasionnellement.

## 2 Optimisation de la métrique

Dans cet article, nous proposons d'*apprendre* la matrice sémantique $\boldsymbol{S}$. D'après (5) et les propriétés d'un produit scalaire, la distance euclidienne au carré entre $x_i$ et $x_j$ vaut dans l'espace transformé par $\boldsymbol{S}$ :

$$\begin{aligned} \tilde{d}_{ij}^2 &\triangleq \|\tilde{\phi}(x_i) - \tilde{\phi}(x_j)\|^2 & (8) \\ &= (\phi(x_i) - \phi(x_j))^T \boldsymbol{S}\boldsymbol{S}^T (\phi(x_i) - \phi(x_j)) & (9) \\ &= \tilde{\kappa}(x_i, x_i) + \tilde{\kappa}(x_j, x_j) - 2\tilde{\kappa}(x_i, x_j). & (10) \end{aligned}$$

Remarquons que $\boldsymbol{S}\boldsymbol{S}^T$ est semi-définie positive ($\boldsymbol{S}\boldsymbol{S}^T \succeq 0$) quelque soit la matrice $\boldsymbol{S}$. La fonction $\tilde{d}_{ij}^2$ est par conséquent une métrique valide (positive, symétrique et satisfaisant l'inégalité triangulaire), pour tout $\boldsymbol{S}$. Pour $\boldsymbol{S}$ la matrice identité en particulier, on retrouve la distance euclidienne

$$d_{ij}^2 = \|\phi(x_i) - \phi(x_j)\|^2 = 2 - 2\kappa(x_i, x_j), \tag{11}$$

car $\tilde{\kappa} = \kappa$ est de norme unité. Supposons maintenant que l'on dispose d'une source d'information contextuelle, présentée sous la forme des ensembles $\mathcal{S}$ et $\mathcal{D}$, où $\mathcal{S}$ (resp. $\mathcal{D}$) représente des paires d'éléments similaires (resp. dissimilaires). Le but est de trouver une transformation $\boldsymbol{S}$ qui caractérise mieux l'information contextuelle, en rapprochant les points similaires et/ou en écartant les points dissimilaires.

### 2.1 Idéalisation de la métrique

Soit $d_{\mathrm{opt}}^2(i,j)$ la distance idéale entre les objets $x_i$ et $x_j$. Bien que l'expression de cette métrique ne soit pas connue, il est possible de calculer sa valeur pour les points de la base d'apprentissage :

$$d_{\mathrm{opt}}^2(i,j) \sim \begin{cases} c & \text{si } (x_i, x_j) \in \mathcal{D} \\ 0 & \text{si } (x_i, x_j) \in \mathcal{S}, \end{cases} \tag{12}$$

où $c > 0$. L'objectif de l'algorithme $\mathrm{DA}_{\mathrm{lign}}$ est d'ajuster $\tilde{d}_{ij}^2$ à son homologue idéal $d_{\mathrm{opt}}^2(i,j)$. Si le calcul explicite de la matrice $\boldsymbol{S}$ est possible pour un espace de faible dimension, ce n'est clairement pas le cas pour les représentations $\phi(x)$ qui nous intéressent. Pour contourner cette difficulté, l'idée consiste à contraindre $\tilde{d}_{ij}^2$ selon une forme paramétrique liée au noyau $\tilde{\kappa}$, ne faisant pas intervenir explicitement $\boldsymbol{S}$.

Dans [5], Wu *et al.* proposent de considérer la contrainte

$$\tilde{\kappa}(x_i, x_j) = \begin{cases} \beta_1 \kappa(x_i, x_j) & \text{si } (x_i, x_j) \in \mathcal{D}, \\ \beta_2 \kappa(x_i, x_j) + (1 - \beta_2) & \text{si } (x_i, x_j) \in \mathcal{S}, \end{cases}$$

où $0 \leq \beta_1, \beta_2 \leq 1$. En utilisant le fait que $\kappa$ est de norme unité et qu'un objet $x_i$ est similaire à lui-même, on trouve $\tilde{\kappa}(x_i, x_i) = 1$. En réinjectant ces expressions dans (10), il est possible d'exprimer $\tilde{d}_{ij}^2$ en fonction de $d_{ij}^2$ :

$$\tilde{d}_{ij}^2 = \begin{cases} \beta_1 d_{ij}^2 + 2(1 - \beta_1) & \text{si } (x_i, x_j) \in \mathcal{D}, \\ \beta_2 d_{ij}^2 & \text{si } (x_i, x_j) \in \mathcal{S}. \end{cases} \tag{13}$$

La distance obtenue est une métrique valide sous réserve que le noyau $\tilde{\kappa}$ soit un noyau valide (semi-défini positif). On montre que cette condition est remplie pour tout $0 \leq \beta_1 \leq \beta_2 \leq 1$ (voir théorème 1, [5]).

Pour $\beta_1 = \beta_2 = 0$, la fonction de distance obtenue correspond à la fonction idéale (12) avec $c = 2$. Pour $\beta_1 = \beta_2 = 1$, on retrouve $d_{ij}^2$. Ces deux cas extrêmes font références respectivement au phénomène de sur/sous-apprentissage. Fixer $\beta_1$ et $\beta_2$ à pour effet de fixer $\boldsymbol{S}$ à une certaine valeur (inconnue). En utilisant la propriété $0 \leq d_{ij}^2 \leq 2$ et la condition $0 \leq \beta_1 \leq \beta_2 \leq 1$, on obtient que $\tilde{d}_{ij}^2 \leq \tilde{d}_{ij}^2$ pour deux points similaires, et $\tilde{d}_{ij}^2 \geq \tilde{d}_{ij}^2$ pour deux points dissimilaires. En d'autres termes, il est possible de décroître la dispersion intra-classe et augmenter la dispersion inter-classe pour les données d'apprentissages, comme souhaité.

### 2.2 Primal $\mathrm{DA}_{\mathrm{lign}}$

L'algorithme $\mathrm{DA}_{\mathrm{lign}}$ formule l'apprentissage de métrique sous forme d'un problème convexe similaire à celui des SVM (Support Vector Machines) [6]. L'objectif est de minimiser une fonction du rang de la matrice $\boldsymbol{S}$, de manière à privilégier les transformations $\boldsymbol{S}$ qui projetent les vecteurs $\phi$ dans un sous-espace de faible dimension. Ce problème étant NP-difficile, il est remplacé en pratique par le problème approximatif visant à minimiser la norme de Frobenius de $\boldsymbol{S}\boldsymbol{S}^T$, $\|\boldsymbol{S}\boldsymbol{S}^T\|_F$. Le problème primal à résoudre est formulé comme suit :

$$\min_{\boldsymbol{S}\boldsymbol{S}^T, \beta_1, \beta_2} \frac{1}{2}\|\boldsymbol{S}\boldsymbol{S}^T\|_F^2 + C_{\mathcal{D}}\beta_1 + C_{\mathcal{S}}\beta_2$$

sous les contraintes :

$$\begin{aligned} &\tilde{d}_{ij}^2 = \beta_1 d_{ij}^2 + 2(1 - \beta_1), && \forall (i,j) \in \mathcal{D}, \\ &\tilde{d}_{ij}^2 = \beta_2 d_{ij}^2, && \forall (i,j) \in \mathcal{S}, \\ &1 - \beta_2 \geq 0, \quad \beta_2 \geq \beta_1, \quad \beta_1 \geq 0. \end{aligned}$$

Comme on l'a noté plus haut, les dernières contraintes sur $\beta_1, \beta_2$ permettent d'imposer implicitement $\boldsymbol{S}\boldsymbol{S}^T \succeq 0$, ce qui justifie la minimisation par rapport à $\boldsymbol{S}\boldsymbol{S}^T$. Cette astuce de calcul permet de réduire considérablement la complexité du problème, tout en garantissant une métrique valide. Dans cette formulation, $C_{\mathcal{D}}$ et $C_{\mathcal{S}}$ sont deux paramètres positifs traduisant un compromis entre la minimisation de $\|\boldsymbol{S}\boldsymbol{S}^T\|_F^2$ et le sur/sous-apprentissage.

### 2.3 Dual $\mathrm{DA}_{\mathbf{lign}}$

Pour résoudre ce problème, l'approche dual consiste à construire son Lagrangien puis à considérer les conditions d'optimalité KKT. Sans le redémontrer ici, on peut se ramener ainsi au problème suivant (voir [5]) :

$$\max_{\boldsymbol{\alpha}} \quad 2\sum_{ij\in\mathcal{D}} \alpha_{ij} + \sum_{ij\in\mathcal{S}}\sum_{k\ell\in\mathcal{D}} \alpha_{ij}\alpha_{kl}H^2_{ij,k\ell} - \frac{1}{2}\Big(\sum_{ij\in\mathcal{D}}\sum_{k\ell\in\mathcal{D}} \alpha_{ij}\alpha_{k\ell}H^2_{ij,k\ell} + \sum_{ij\in\mathcal{S}}\sum_{k\ell\in\mathcal{S}} \alpha_{ij}\alpha_{k\ell}H^2_{ij,k\ell}\Big),$$

sous les contraintes :

$$\begin{aligned} C_{\mathcal{S}} &= 2\sum_{ij\in\mathcal{S}} \alpha_{ij}\big(1-\kappa(x_i,x_j)\big), \\ C_{\mathcal{D}} &\geq 2\sum_{ij\in\mathcal{D}} \alpha_{ij}\kappa(x_i,x_j), \\ \alpha_{ij} &\geq 0, \end{aligned}$$

où $H_{ij,k\ell} = \kappa(x_i,x_k) - \kappa(x_i,x_\ell) - \kappa(x_j,x_k) + \kappa(x_j,x_\ell)$.

Le problème à résoudre est une problème d'optimisation quadratique, sous contraintes linéaires. Sa résolution peut s'effectuer par programmation quadratique. Les expressions du noyau et de la distance associée, pour des individus tests, sont données en annexe.

## 3 Résultats expérimentaux

Pour la partie expérimentale, nous avons utilisé la base de données TechTC300[1]. Nous présentons dans cette version papier uniquement quelques résultats obtenus avec l'ensemble d'apprentissage '1092 vs 1110'. Ces données sont constituées de 269 documents HTML provenant de différents sites Internet et répartis en deux classes, voir [9] pour des détails sur la méthode d'étiquettage employée.

Les expérimentations ont été menées sous MATLAB. Le classifieur était de type $l_1$-SVM. Les ensembles $\mathcal{S}$ et $\mathcal{D}$ ont été construits en sélectionnant aléatoirement $m$ paires de documents $(x_i, x_j)$, puis en affectant les indices $(i,j)$ à $\mathcal{S}$ pour $(x_i, x_j)$ de même classe, à $\mathcal{D}$ sinon. Notons ici que le nombre de paires distinctes possibles est $n(n-1)/2$, où $n$ est la taille de l'ensemble d'apprentissage. Afin que le problème dual ne dépasse pas la complexité du problème SVM à résoudre ensuite, nous avons choisi $m = n$. L'erreur de généralisation a été estimée par validation croisée (méthode des 5-folds). Le tableau 1 présente les résultats obtenus par apprentissage de métrique en fonction des paramètres $C_{\mathcal{S}}$ et $C_{\mathcal{D}}$. Le tableau 2 compare ces résultats aux approches courantes. On remarque que l'opération de normalisation améliore les performances du noyau *sac de mots* non modifié et TF-IDF. On constate également que le noyau appris par la méthode proposée est celui qui offre les meilleures performances. En outre, celles-ci se sont avérées en moyennes, pour les valeurs de $C_{\mathcal{S}}, C_{\mathcal{D}}$ considérées, toujours meilleures que les noyaux non modifié, normalisé et TF-IFD.

[1] http ://techtc.cs.technion.ac.il/techtc.html

## 4 Conclusion

Dans cet article, nous avons appliqué l'apprentissage de métrique à l'optimisation de la matrice sémantique d'un classifieur textuel. Etant donné une source d'information contextuelle, formée de paires de documents similaires et dissimilaire, l'algorithme $\mathrm{DA}_{\mathrm{lign}}$ employé permet de déterminer la matrice sémantique optimale, sans qu'il soit nécessaire de l'exhiber. Nous avons montré empiriquement l'intérêt de cet algorithme dans applications de classification par machines à vecteurs supports.

## Références


[1] G. Salton, A. Wang, and C. Yang, "A vector space model for information retrieval," *Journal of the American Society of Information Science*, vol. 18, pp. 613–620, 1975.

[2] E. Leopold and J. Kinderman, "Text categorization with Support Vector Machines. How to represent texts in input space ?" *Machine Learning : Special Issue on Support Vector and Kernel Methods*, vol. 46, pp. 423–44, 2002.

[3] S. K. M. Wong, W. Ziarko, and P. C. N. Wong, "Generalized vector space model in information retrieval," *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 18–25, 1985.

[4] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society of Information Science*, vol. 41, pp. 391–407, 1990.

[5] G. Wu, N. Panda, and E. Y. Change, "Formulating distance functions via the kernel trick," in *ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2005, pp. 703–709.

[6] V. N. Vapnik, *Statistical Learning Theory.* New York : Wiley, 1998.

[7] T. Joachims, *Learning to classify text using Support Vector Machines.* Kluwer, 2002.

[8] J. T. Kwok and I. W. Tsang, "Learning with idealized kernels," *Proc. of the Twentieth International Conference on Machine Learning*, pp. 400–407, 2003.

[9] D. Davidov, E. Gabrilovich, and S. Markovitch, "Parameterized generation of labeled datasets for text categorization based on hierarchical directory," *The 27th Annual International ACM SIGIR Conference*, pp. 250–257, 2004.

# 5 Annexes

## 5.1 Tableaux de résultats

| $C_{\mathcal{S}} \setminus C_{\mathcal{D}}$ | 0.2 | 0.5 | 1 | 2 | 5 | 10 | 20 | 100 | 200 |
|---|---|---|---|---|---|---|---|---|---|
| 0.2 | 6.69 | 6.69 | 7.08 | 7.08 | 5.58 | 5.95 | 5.58 | 7.05 | 5.95 |
| 0.5 | 8.18 | 6.31 | 5.59 | 6.69 | 6.71 | 6.31 | 5.22 | 6.33 | 7.81 |
| 1 | 7.78 | 6.70 | 8.18 | 6.70 | 6.68 | 6.31 | 4.10 | 6.32 | 7.43 |
| 2 | 6.68 | 5.22 | 4.10 | 7.06 | 3.34 | 3.34 | 7.06 | 5.95 | 8.53 |
| 5 | 8.55 | 6.72 | 5.59 | 6.32 | 4.81 | 7.06 | 3.72 | 6.31 | 7.06 |
| 10 | 7.04 | 5.21 | 7.42 | 9.70 | 5.58 | 4.10 | 4.84 | 5.59 | 8.53 |
| 20 | 7.81 | 7.81 | 7.43 | 9.66 | 6.32 | 8.53 | 8.83 | 7.06 | 8.92 |
| 100 | 10.4 | 7.47 | 9.32 | 10.03 | 6.33 | 5.95 | 4.84 | 7.42 | 6.69 |
| 200 | 10.4 | 8.93 | 10.43 | 10.06 | 8.18 | 7.45 | 10.03 | 7.81 | 10.05 |

TAB. 1 – Erreur de généralisation estimée en fonction de $(C_{\mathcal{S}}, C_{\mathcal{D}})$.

| noyau | erreur |
|---|---|
| sac de mots non modifié | 13.39% |
| sac de mots + normalisation | 11.49% |
| TF-IFD | 11.15% |
| TF-IFD + normalisation | 6.69% |
| noyau appris | 3.34% |

TAB. 2 – Erreur de généralisation estimée.

## 5.2 Expression du noyau appris pour des individus hors ensemble d'apprentissage

D'après les conditions d'optimalité du problème dual, la solution $\boldsymbol{S}\boldsymbol{S}^T$ du problème primal s'exprime (voir [5]) :

$$\boldsymbol{S}\boldsymbol{S}^T = \sum_{ij\in\mathcal{D}} \alpha_{ij}\big(\phi(x_i)-\phi(x_j)\big)\big(\phi(x_i)-\phi(x_j)\big)^T - \sum_{ij\in\mathcal{S}} \alpha_{ij}\big(\phi(x_i)-\phi(x_j)\big)\big(\phi(x_i)-\phi(x_j)\big)^T.$$

Afin de simplifier les notations, notons $\phi_i = \phi(x_i)$ et $\kappa_{xx'} = \kappa(x,x')$. Réinjecter l'expression de $\boldsymbol{S}\boldsymbol{S}^T$ dans (5) donne :

$$\begin{aligned}
\tilde{\kappa}(x,x') &= \phi(x)^T \boldsymbol{S}\boldsymbol{S}^T \phi(x') \\
&= \phi(x)^T \Bigg( \sum_{ij\in\mathcal{D}} \alpha_{ij}\big(\phi_i-\phi_j\big)\big(\phi_i-\phi_j\big)^T - \sum_{ij\in\mathcal{S}} \alpha_{ij}\big(\phi_i-\phi_j\big)\big(\phi_i-\phi_j\big)^T \Bigg)\phi(x') \\
&= \sum_{ij\in\mathcal{D}} \alpha_{ij}\phi(x)^T \Big( \big(\phi_i-\phi_j\big)\big(\phi_i-\phi_j\big)^T \Big)\phi(x') - \sum_{ij\in\mathcal{S}} \alpha_{ij}\phi(x)^T \Big( \big(\phi_i-\phi_j\big)\big(\phi_i-\phi_j\big)^T \Big)\phi(x') \\
&= \sum_{ij\in\mathcal{D}} \alpha_{ij} \Big( \phi(x)^T(\phi_i-\phi_j) \Big) \Big( (\phi_i-\phi_j)^T\phi(x') \Big) - \sum_{ij\in\mathcal{S}} \alpha_{ij} \Big( \phi(x)^T(\phi_i-\phi_j) \Big) \Big( (\phi_i-\phi_j)^T\phi(x') \Big) \\
&= \sum_{ij\in\mathcal{D}} \alpha_{ij} \Big( \phi(x)^T(\phi_i-\phi_j) \Big) \Big( (\phi_i-\phi_j)^T\phi(x') \Big) - \sum_{ij\in\mathcal{S}} \alpha_{ij} \Big( \phi(x)^T(\phi_i-\phi_j) \Big) \Big( (\phi_i-\phi_j)^T\phi(x') \Big) \\
&= \sum_{ij\in\mathcal{D}} \alpha_{ij}\big(\kappa_{xx_i}-\kappa_{xx_j}\big)\big(\kappa_{x_ix'}-\kappa_{x_jx'}\big) - \sum_{ij\in\mathcal{S}} \alpha_{ij}\big(\kappa_{xx_i}-\kappa_{xx_j}\big)\big(\kappa_{x_ix'}-\kappa_{x_jx'}\big),
\end{aligned}$$

que l'on peut calculer sans expliciter $\phi$ grâce à l'astuce des noyaux. Par des calculs similaires, on montre sans difficultés :

$$\begin{aligned}
\tilde{d}(x,x')^2 &= (\phi(x)-\phi(x'))^T \boldsymbol{S}\boldsymbol{S}^T(\phi(x)-\phi(x')) \\
&= \sum_{ij\in\mathcal{D}} \alpha_{ij}\big(\kappa_{xx_i}-\kappa_{xx_j}-\kappa_{x_ix'}+\kappa_{x_ix'}\big)^2 - \sum_{ij\in\mathcal{S}} \alpha_{ij}\big(\kappa_{xx_i}-\kappa_{xx_j}-\kappa_{x_ix'}+\kappa_{x_ix'}\big)^2.
\end{aligned}$$