Sélection du noyau

Sommaire

La question du choix du noyau est un point essentiel de l'apprentissage par Machines à Vecteurs de Support. En effet, la fonction noyau détermine le champ des surfaces de décision possibles et, comme nous l'avons vu, elle implique l'utilisation d'une technique de classification sous-jacente. Nous donnerons dans cette section une vue d'ensemble des possibilités qu'offrent les noyaux et présenterons diverses stratégies d'affinage ou de sélection de noyau pour une tâche donnée. La plupart des noyaux, excepté le noyau linéaire, incluent un ou plusieurs paramètres dans leur expression, appelés *hyper-paramètres*. Nous traiterons en section 4.2 des stratégies de détermination des valeurs optimales de ces paramètres. Un état de l'art des critères les plus courants pour estimer l'erreur *Leave-one-out*, en section 4.3, nous permettra d'introduire les notions nécessaires et de comparer ces derniers à deux récents critères de la littérature, basés sur la maximisation de la séparabilité des classes, dont nous proposons dans la section 4.4 l'application pour la première fois dans le domaine de l'indexation audio.

Nous porterons en outre une attention au facteur d'erreur C, qui fixe le compromis entre la pénalisation des erreurs et la minimisation de l'erreur de généralisation dans les Machines à Marge Souple. Nous aborderons par la suite les aspects propres à ce paramètre en section 4.5, que nous exploiterons pour proposer des améliorations sur les critères de séparabilité des classes introduits.

4.1 Illustration sur des données artificielles

Nous allons montrer l'importance de l'affinage des hyper-paramètres à travers une courte étude sur des données artificielles à deux dimensions. Cette étude portera sur le paramètre σ du noyau RBF gaussien, parce que les noyaux sigmoïdes sont très instables (le noyau n'est pas positif pour toutes les valeurs de ses paramètres), et les noyaux polynomiaux ont intérêt limité sur deux dimensions. On s'intéressera donc à des problèmes définis sur des variables bi-dimensionnelles $\boldsymbol{x} = [x_1, x_2]$. Dans le cas de données artificielles séparables, on pourra définir une fonction de décision *idéale* $f_I(\boldsymbol{x})$ telle que $y = f_I(\boldsymbol{x})\ \forall \boldsymbol{x}$, où y est la classe associée à l'exemple $\boldsymbol{x}$.

4.1.1 Surface de décision

Le problème de l'échiquier constitue un très bon exemple de données artificielles linéairement inséparables. Si l'on fixe à N_C le nombre de case par côté, la fonction de décision idéale $f_I : \mathbb{R}^2 \rightarrow [0;1] \times [0;1]$ est définie comme suit :

$$f_I(x_1, x_2) = \text{sign}(\text{mod}(\lfloor N_C(x_1 + x_2) \rfloor, 2)),$$

où $\text{mod}(a, b)$ est le reste de la division entière de a par b, et l'opérateur $\lfloor x \rfloor$ désigne l'arrondi par défaut. La figure 4.1 montre la fonction de décision idéale pour un échiquier à $N_C = 3$ cases de largeur, ainsi que 250 exemples générés aléatoirement. Les croix claires désignent les exemples de la classe 1, symbolisée par les régions noires pour la fonction de décision, tandis que les cercles foncés désignent les exemples de la classe 2, symbolisée par les régions blanches pour la fonction de décision.

Les figures 4.2(a), (b) et (c) illustrent le résultat de l'apprentissage par SVM avec noyau gaussien RBF pour différentes valeurs de σ. La ligne pleine blanche représente la surface de séparation et les pointillés gris représentent les surfaces de marge pour chaque classe. Les Vecteurs de Support sont donc les exemples situés sur ces lignes grises.

Si l'on rappelle l'expression du noyau gaussien RBF,

$$k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{y}||^2}{d\,\sigma^2}\right),$$

celui-ci traduit bien une mesure de similarité basée sur la distance entre les exemples. Lorsque σ augmente, le terme sur lequel s'applique l'exponentielle inverse tend vers 0, ce qui accroît la similarité entre les exemples. À l'inverse, lorsque l'on réduit le terme σ, la distance entre les exemples se trouve amplifiée.

Ce phénomène apparaît clairement dans l'exemple présent. Lorsque σ est trop bas (figure 4.2(a)), les exemples sont plus distants entre eux et la fonction de décision doit donc inclure plus de Vecteurs de Support pour pouvoir couvrir tout l'espace. On voit en effet sur la figure que de

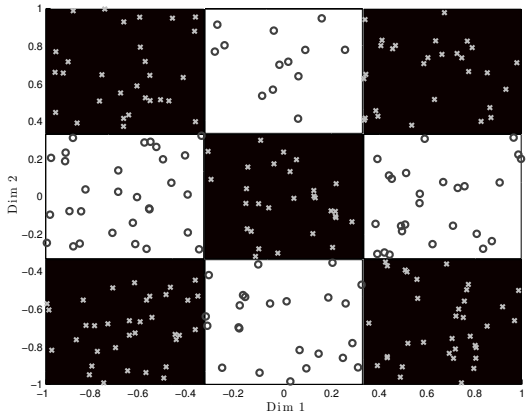
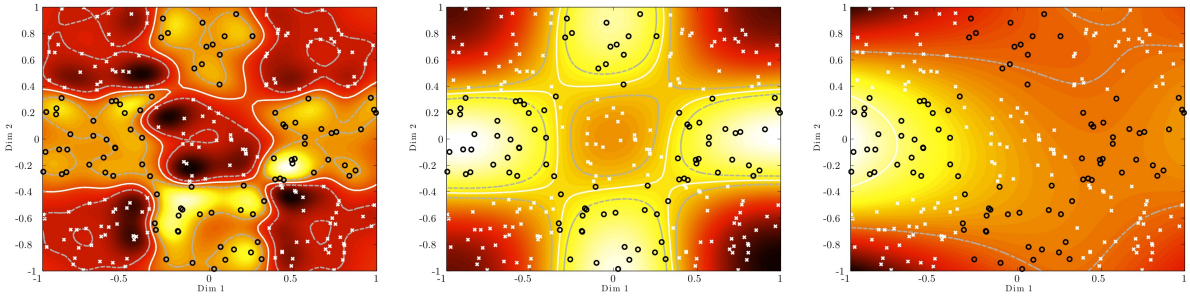


FIGURE 4.1 – Fonction de décision idéale de la distribution *Echiquier* et répartition des 100 exemples d'apprentissage.



(a) $\sigma = 0.2, C = 100$ (b) $\sigma = 1, C = 100$ (c) $\sigma = 3, C = 100$

FIGURE 4.2 – Surfaces de décision obtenues par apprentissage SVM à noyau gaussien RBF pour différentes valeurs de σ sur la distribution *Echiquier*.

nombreux exemples figurent sur la surface de marge, qui se trouve ainsi trop adaptée à l'ensemble d'apprentissage, au détriment de la capacité de généralisation du classificateur. On parle alors de sur-apprentissage (ou d'*overfitting*).

Lorsque, par contre, σ est trop élevé (figure 4.2(c)), la mesure de similarité ne permet plus de distinguer les exemples de classes opposées. Ainsi l'algorithme est incapable de déduire une surface de décision traduisant la frontière entre les classes.

La figure centrale (4.2(b)) présente dans ce cas un bon compromis entre généralisation et erreur de classification.

4.1.2 Tolérance aux *outliers*

On utilise ici une distribution plus simple, nommée *Courbe* et représentée figure 4.3, pour illustrer l'influence du paramètre C par rapport à la présence d'*outliers*. 100 exemples sont générés aléatoirement, dont 8 *outliers* sont assignés à la classe erronée. La figure 4.4 montre les résultats de l'apprentissage par SVM avec noyau gaussien RBF pour différentes combinaisons de valeurs pour σ et C.

On peut ainsi constater qu'une valeur trop basse du facteur d'erreur C (figures (a) à gauche) induit une trop grande tolérance aux erreurs de classification. Ainsi le classificateur peine à établir une surface de décision qui réponde au problème posé puisque les exemples peuvent se situer indifféremment d'un côté ou de l'autre de la surface de décision.

A l'inverse, une valeur trop élevée de C (figures (c) à droite) pousse le classificateur à prendre en compte le moindre exemple erroné (dans la mesure où le paramètre σ lui permet de complexifier la surface de décision, ce qui n'est pas le cas par exemple pour la figure (c,3) où $\sigma = 6$). La comparaison des figures (c,1) et (c,2) illustre clairement la tendance à « encercler » chaque *outlier*

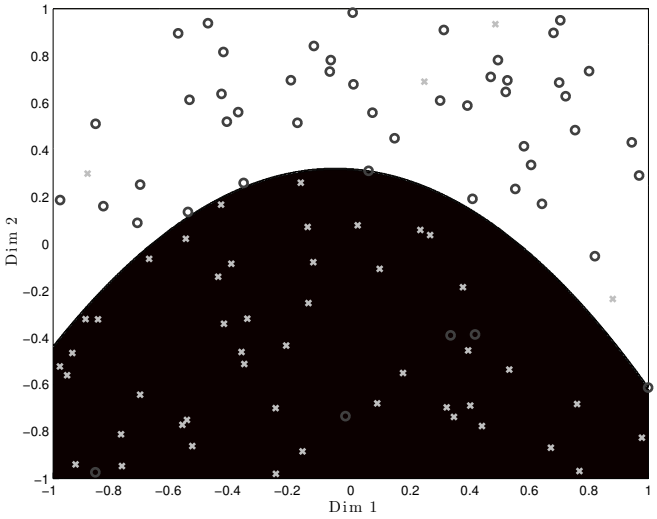


FIGURE 4.3 – Fonction de décision idéale de la distribution *Courbe* et répartition des 100 exemples d'apprentissage.

pour C trop élevé.

Dans le cas étudié, $C = 2$ (figures (b) au centre) constitue un bon compromis. Néanmoins, on voit que cette valeur n'a de sens que pour un σ adéquat. Les paramètres sont donc fortement liés entre eux, ce qui explique la complexité de l'affinage des noyaux puisque les paramètres ne peuvent être affinés indépendamment.

Nous présentons dans la section suivante les stratégies possibles pour l'affinage des paramètres, ainsi que les critères employés. On reviendra plus précisément dans la section 4.5 sur la question du facteur d'erreur C, et sa relation au paramètre σ.

4.2 Stratégies d'affinage

On aborde ici le problème de l'affinage des hyper-paramètres indépendamment de leur nature et de leur nombre. On considère donc qu'un noyau $k_{\mathbf{\Theta}}$ est paramétré par P valeurs $\mathbf{\Theta} = [\theta_1, \ldots, \theta_P]$.

4.2.1 Recherche par maillage

La recherche par maillage (*grid-search*) est la méthode la plus généralement employée. Elle consiste à évaluer les performances du classifieur SVM appris sur un ensemble fini de V valeurs $\mathcal{V} = \{\mathbf{\Theta}_i, \quad i \in [1, \ldots, V]\}$. Soit $\mathcal{P}(k_{\mathbf{\Theta}})$ la mesure de performances du noyau $k_{\mathbf{\Theta}}$, l'algorithme consiste donc à retenir la valeur $\hat{\mathbf{\Theta}}$ telle que

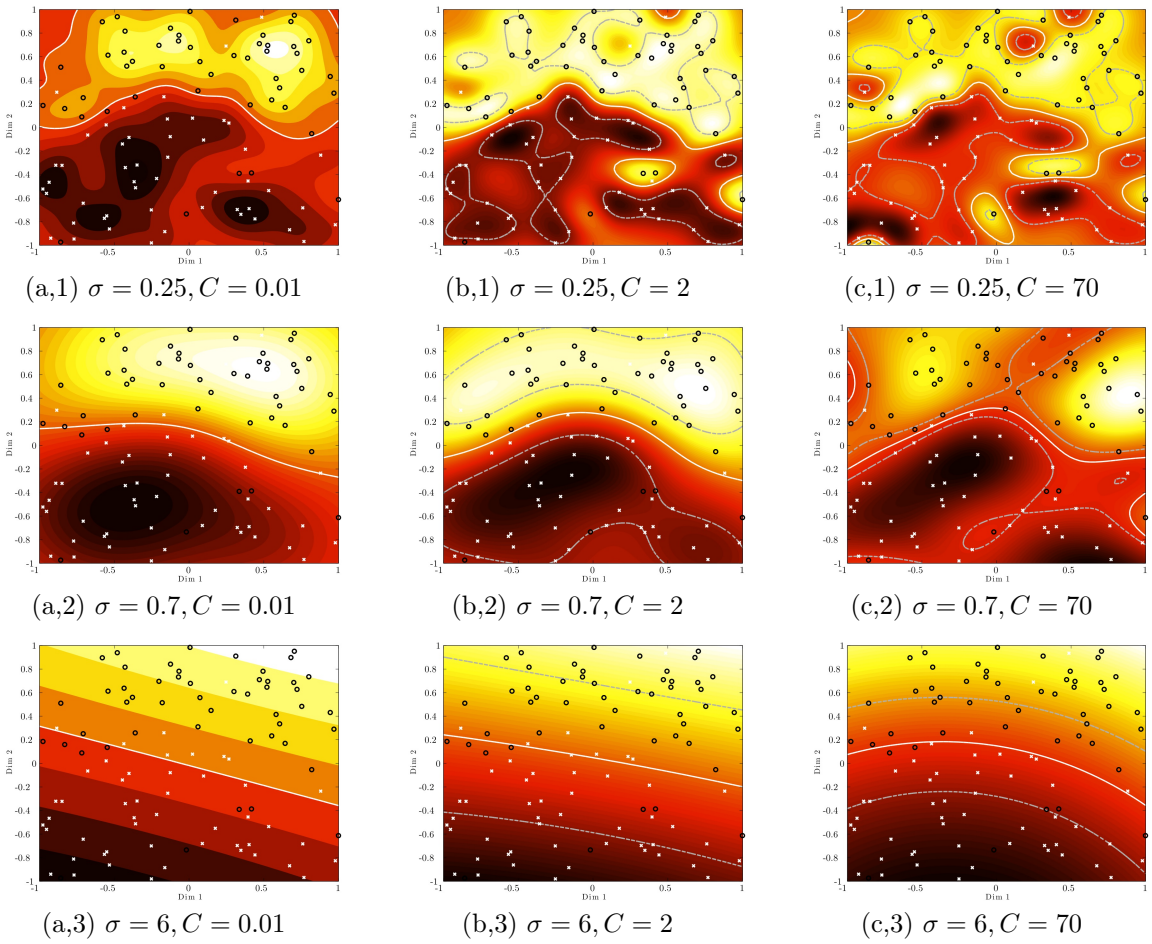
$$\hat{\mathbf{\Theta}} = \arg\max_{\mathbf{\Theta} \in \mathcal{V}} \mathcal{P}(k_{\mathbf{\Theta}}).$$

Concernant le choix des valeurs de l'ensemble $\mathcal{V}$, on utilise généralement pour chaque paramètre un ensemble de valeurs également réparties dans un intervalle donné. $\mathcal{V}$ est alors le produit cartésien de ces ensembles et constitue un *maillage* de l'espace des paramètres sur un intervalle donné. Il est également courant d'utiliser des valeurs logarithmiquement réparties.

Nous détaillerons par la suite (section 4.3) la plupart des critères permettant d'évaluer la mesure de performance d'un classifieur SVM. La solution la plus basique et la plus généralement employée consiste à mesurer le taux d'erreur sur un ensemble dit de *validation*.

La recherche par maillage souffre ainsi de deux défauts majeurs :

- La complexité algorithmique est polynomiale, en $O(n^P)$. On fait donc face à une explosion combinatoire dès que le nombre de paramètres dépasse 1 ou 2.



(a,1) $\sigma = 0.25, C = 0.01$ (b,1) $\sigma = 0.25, C = 2$ (c,1) $\sigma = 0.25, C = 70$

(a,2) $\sigma = 0.7, C = 0.01$ (b,2) $\sigma = 0.7, C = 2$ (c,2) $\sigma = 0.7, C = 70$

(a,3) $\sigma = 6, C = 0.01$ (b,3) $\sigma = 6, C = 2$ (c,3) $\sigma = 6, C = 70$

FIGURE 4.4 – Surfaces de décision obtenues par apprentissage SVM à noyau gaussien RBF pour différentes valeurs de σ et C sur la distribution *Courbe*.

- En supposant que le critère de performance est fiable, on n'a aucune garantie que le maximum global (ou qu'un maximum local) se trouve dans le champ du maillage. Un maillage suffisamment serré couplé à un critère régulier résoud ce problème, mais au prix possible d'une explosion combinatoire en présence de nombreux paramètres.

4.2.2 Optimisation

Une autre stratégie consiste, plutôt que de tenter de couvrir l'espace de recherche, à évaluer itérativement la valeur optimale par mises à jour successives en fonction d'un critère de performance. Cette approche sollicite donc des méthodes d'optimisation, domaine très large couvrant le problème de la recherche d'extrema. Les méthodes de programmation mathématique (linéaire, quadratique, etc...) ne peuvent être utilisées ici puisque le problème n'est pas exprimé analytiquement, tandis que les algorithmes dérivés de la descente de gradient sont applicables. De nombreuses approches de ce type existent dans la littérature [26], parmi lesquelles la Méthode de Newton, la descente de gradient stochastique ou le gradient conjugué. Néanmoins, une étude des méthodes d'optimisation sort du domaine de cette thèse et nous nous restreindrons à l'usage de la plus simple, la descente (ou montée) de gradient. On peut résumer celle-ci par l'algorithme 1 présenté ci-dessous.

La condition d'arrêt peut également s'appliquer à la différence de performance entre deux itérations. La descente de gradient implique de pouvoir calculer le gradient de la mesure de performance par rapport aux paramètres du noyau. C'est là la principale restriction liée à cette méthode.

Algorithme 1 Optimisation

$k_{\boldsymbol{\Theta}}$ le noyau paramétré par ...
$\boldsymbol{\Theta}$, de valeur initiale $\boldsymbol{\Theta}_0$.
$\mathcal{P}(k_{\boldsymbol{\Theta}})$ mesure de performance sur le noyau $k_{\boldsymbol{\Theta}}$
λ pas d'avancement
ϵ paramètre d'arrêt

répéter
  Estimer le gradient $\Delta_{\boldsymbol{\Theta}}\mathcal{P}(k_{\boldsymbol{\Theta}})$
  $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} + \lambda \cdot \Delta_{\boldsymbol{\Theta}}\mathcal{P}(k_{\boldsymbol{\Theta}})$
jusqu'à $\Delta_{\boldsymbol{\Theta}}\mathcal{P}(k_{\boldsymbol{\Theta}}) \leq \epsilon$

4.2.3 Recherche par voisinage

Momma et Bennett proposent [161] pour l'affinage de paramètres une recherche par pas successifs qui constitue un compromis intéressant entre optimisation et recherche par maillage pour le cas de critères non dérivables. A chaque itération k le critère est évalué sur des points voisins de la valeur actuelle, dont la disposition forme ce que les auteurs appellent un *pattern*, et le point actuel est déplacé au point voisin minimisant le critère, jusqu'à convergence. Le *pattern* le plus simple consiste à sélectionner les points à une distance Δ_k, décroissante, pour chacune des directions axiales de l'espace.

Ce critère permet de s'affranchir de la contrainte de dérivabilité et restreint fortement l'espace de recherche par rapport à la recherche par maillage. Toutefois, il reste très coûteux lorsque l'espace est de dimension élevée, et n'offre aucune garantie quant à la globalité du maximum trouvé.

En pratique on préférera se limiter ici à des critères dérivables permettant l'usage de la descente de gradient.

4.3 Critères d'évaluation basés sur l'erreur de généralisation

Nous présentons dans cette section la plupart des critères couramment utilisés par l'évaluation des noyaux. Ceux-ci consistent en général en l'expression d'une borne supérieure sur l'erreur *Leave-One-Out*. On trouvera dans [47], [90] et [66] une vue d'ensemble assez didactique des critères qui suivent.

4.3.1 Erreur sur un ensemble de validation

Si l'on dispose de suffisamment de données, la solution la plus simple consiste à estimer le taux d'erreur d'une machine SVM sur un ensemble de validation, dont les exemples sont distincts de ceux de l'ensemble d'apprentissage. Soit un ensemble de validation de p éléments $\mathcal{S}_V = \{(\boldsymbol{x}'_i, y'_i)\}_{i\in[1,\dots,p]}$, on définit l'erreur par :

$$\mathcal{P}_{val} = \frac{1}{p}\sum_{i=1}^{p} H\left(-y'_i\, f(\boldsymbol{x}'_i)\right),$$

où H est la fonction de Heaviside ($H(x) = 1$ si $x > 0$, et $H(x) = 0$ sinon).

Ce critère est simple mais il restreint le volume de données d'apprentissage et suppose que l'ensemble de validation est caractérisé par la même distribution sous-jacente, ce qui peut être difficile à valider sur des données réelles. On a donc peu de garanties sur l'absence de biais du critère.

4.3.2 Validation croisée

La validation croisée est une variante plus robuste du critère précédent. Les exemples d'un ensemble de base sont partagés en k sous-ensembles répartis aléatoirement. Chacun des k sous-ensembles est utilisé comme ensemble de validation pour calculer l'erreur d'un classifieur appris

sur l'union des $k-1$ autres sous-ensembles. Le critère de validation croisée est la moyenne des erreurs obtenues sur les k itérations.

Cette démarche permet ainsi de réduire le biais possible entre les ensembles d'apprentissage et de validation, mais accroît le temps de calcul d'un facteur k.

4.3.3 Erreur *Leave-One-Out*

L'erreur *Leave-One-Out* (ou LOO), littéralement « un laissé dehors », peut être vue comme une validation croisée poussée à l'extrême. Elle consiste à évaluer le taux d'erreur en classifiant chaque exemple $\boldsymbol{x}_i$ de l'ensemble $\mathcal{S}$ par le classifieur SVM appris sur l'ensemble $\mathcal{S}^{\setminus \boldsymbol{x}_i}$ comprenant tous les autres exemples, soit :

$$\mathcal{P}_{LOO} = \frac{1}{n} \sum_{i=1}^{n} H\left(-y_i\, f^i(\boldsymbol{x}_i)\right),$$

où f^i est la fonction de décision du classifieur SVM appris sur l'ensemble $\mathcal{S}^{\setminus \boldsymbol{x}_i}$.

L'estimation de l'erreur LOO est connue pour être presque non biaisée [142] (le presque faisant référence au fait que l'erreur porte sur $n-1$ échantillons au lieu de n) mais celui-ci est extrêmement coûteux puisqu'il nécessite à priori l'apprentissage de n classifieurs. Si l'on appelle f^0 le classifieur appris sur tous les exemples, on a :

$$\begin{aligned} \mathcal{P}_{LOO} &= \frac{1}{n} \sum_{i=1}^{n} H\left(-y_i\, f^0(\boldsymbol{x}_i) + y_i \left[f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i)\right]\right) && (4.1) \\ &= \frac{1}{n} \operatorname{Card}\left\{i, y_i \left[f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i)\right] > y_i\, f^0(\boldsymbol{x}_i)\right\}. && (4.2) \end{aligned}$$

Cette expression permet, dans le cas des SVM, de définir une borne supérieure à l'erreur LOO en bornant l'expression $y_i \left[f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i)\right]$. On peut remarquer que l'exclusion d'un vecteur non-support de l'ensemble d'aprentissage ne modifie pas la fonction de décision, d'où $f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i) = 0$ pour $\boldsymbol{x}_i \notin \mathcal{S}_{SV}$ (où $\mathcal{S}_{SV}$ est l'ensemble des vecteurs de support pour le classifieur f^0). Seul l'apprentissage des classifieurs f^i avec $\boldsymbol{x}_i \in \mathcal{S}_{SV}$ est donc nécessaire pour calculer l'erreur *Leave-One-Out*.

Malgré ce constat, le calcul de l'erreur LOO demeure prohibitif. De nombreuses méthodes [119][64][173][110] ont été suggérées qui permettent d'en alléger le calcul, généralement en bornant à l'excès l'expression $y_i \left[f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i)\right]$. Nous détaillons par la suite quelques-unes des plus courantes.

4.3.4 Nombre de Vecteurs de Support

Dans le cas de SVM à marge dure, le premier terme de la fonction de Heaviside dans l'équation 4.1 est borné supérieurement puisque $y_i\, f^0(\boldsymbol{x}_i) \geq 1$. La fonction de Heaviside étant croissante monotone, on a donc :

$$\mathcal{P}_{LOO} \leq \frac{1}{n} \sum_{i=1}^{n} H\left(y_i \left[f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i)\right] - 1\right).$$

Nous avons vu que $f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i) = 0$ pour tout vecteur « non support », on peut donc restreindre la somme précédente aux vecteurs de support :

$$\mathcal{P}_{LOO} \leq \frac{1}{n} \sum_{i/\boldsymbol{x}_i \in \mathcal{S}_{SV}} H\left(y_i \left[f^0(\boldsymbol{x}_i) - f^i(\boldsymbol{x}_i)\right] - 1\right).$$

On peut ainsi approximer grossièrement l'erreur LOO en bornant chaque fonction de Heaviside par 1, ce qui donne l'estimée $\mathcal{P}_{NSV}$ (pour *Number of Support Vectors*) :

$$\mathcal{P}_{LOO} \leq \mathcal{P}_{NSV} = \frac{\operatorname{Card} \mathcal{S}_{SV}}{n},$$

où $\operatorname{Card} \mathcal{S}_{SV}$ est le nombre de vecteurs de support. Cette estimation de l'erreur est très simple dans sa formulation mais discontinue par rapport aux paramètres, ce qui empêche l'application de méthodes d'optimisation usuelles sur ce critère.

4.3.5 Estimée $\xi\alpha$

Joachims [115] fournit une borne supérieure à l'erreur LOO ne dépendant que des variables calculées durant l'apprentissage des SVM. Il montre en effet que

$$\mathcal{P}_{LOO} \leq \mathcal{P}_{\xi\alpha} = \frac{1}{n} \operatorname{Card} \left\{ i, 2\alpha_i R^2 + \xi_i \geq 1 \right\},$$

où R est une estimation du rayon minimal de la sphère contenant tous les exemples dans l'espace transformé (on trouvera dans l'annexe A les différentes techniques d'estimation du rayon R). Les grandeurs α_i et ξ_i sont respectivement les facteurs de Lagrange et les variables d'écarts définis dans le problème d'optimisation des SVM à marge souple ; leur calcul n'induit pratiquement aucun coût supplémentaire après apprentissage d'un SVM. Toutefois, comme pour $\mathcal{P}_{NSV}$, ce critère est un dénombrement et n'est pas dérivable.

4.3.6 Borne Rayon-Marge

Il est également montré [230] que dans le cas de données séparables, l'erreur de généralisation (qui est estimée de manière presque non-biaisée par l'erreur LOO) est bornée par la valeur suivante :

$$\mathcal{P}_{RM} = \frac{1}{n} \frac{R^2}{M^2} = \frac{1}{n} R^2 \|\boldsymbol{w}\|^2,$$

où R est le rayon défini précédemment, M est la marge du classifieur SVM, et $\boldsymbol{w}$ le vecteur normal de l'hyperplan de séparation. Cette borne est déduite d'une majoration par x de la fonction de Heaviside $H(x-1)$, qui permet de supprimer les discontinuités. Le critère $\mathcal{P}_{RM}$ appelé borne Rayon-Marge (*Radius-Margin bound*) est donc dérivable. Soit un paramètre à P composantes $\boldsymbol{\Theta} = [\theta_1 \dots \theta_P]$,

$$\frac{\partial R^2 \|\boldsymbol{w}\|^2}{\partial \boldsymbol{\Theta}} = R^2 \frac{\partial \|\boldsymbol{w}\|^2}{\partial \boldsymbol{\Theta}} + \|\boldsymbol{w}\|^2 \frac{\partial R^2}{\partial \boldsymbol{\Theta}}.$$

On déduit immédiatement de l'expression de $\boldsymbol{w}$ (éq. 3.31), celle de la dérivée de $\|\boldsymbol{w}\|^2$:

$$\frac{\partial \|\boldsymbol{w}\|^2}{\partial \boldsymbol{\Theta}} = -\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \frac{\partial k(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{\Theta}}.$$

L'estimation du rayon R est traitée dans l'annexe A. On suppose que celui-ci peut s'exprimer en fonction des exemples, par la combinaison linéaire suivante :

$$R^2 = \sum_{i=1}^{n} \beta_i k(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{i,j=1}^{n} \beta_i \beta_j k(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{4.3}$$

On en déduit :

$$\frac{\partial R^2}{\partial \boldsymbol{\Theta}} = \sum_{i=1}^{n} \beta_i \frac{\partial k(\boldsymbol{x}_i, \boldsymbol{x}_i)}{\partial \boldsymbol{\Theta}} - \sum_{i,j=1}^{n} \beta_i \beta_j \frac{\partial k(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{\Theta}}.$$

Ce critère est très utilisé dans la littérature et permet d'affiner les paramètres avec précision. Cependant la majoration de la fonction de Heaviside par x accentue la contribution des erreurs importantes. De plus, on rappelle que même si le critère se comporte bien de façon générale il repose sur la supposition que les données d'apprentissage sont séparables, ce qui remet en question sa pertinence théorique dans le cas de données non séparables (ce qui est généralement le cas).

4.3.7 Borne sur l'étendue

Vapnik et Chapelle [46] [231] définissent le concept d'*étendue* (*span*) d'un vecteur support $\boldsymbol{x}_p$, comme la distance (dans l'espace transformé) entre celui-ci et l'espace Λ_p :

$$S_p = d(\Phi(\boldsymbol{x}_p), \Lambda_p) = \min_{\boldsymbol{x} \in \Lambda_p} (\Phi(\boldsymbol{x}_p), \Phi(\boldsymbol{x})),$$

où Λ_p est défini comme un sous ensemble de l'espace engendré par les autres vecteurs de support, délimité par une contrainte additionnelle :

$$\Lambda_p = \left\{ \sum_{i \neq p, \alpha_i > 0} \lambda_i \Phi(\boldsymbol{x}_i), \quad \sum_{i \neq p} \lambda_i = 1 \right\}.$$

Le *span* est ainsi une mesure de la distance entre un vecteur support et les autres. Intuitivement, plus cette mesure est réduite, et moins la procédure de *Leave-One-Out* sur cet exemple est susceptible de produire une erreur. On peut montrer [231] que si l'ensemble des vecteurs de support ne change pas entre les classifieurs f^0 et f^p (on reprend ici les notations de la section 4.3.3), alors

$$y_p \left(f^0(\boldsymbol{x}_p) - f^p(\boldsymbol{x}_p) \right) = \alpha_p S_p^2.$$

La borne sur l'étendue se déduit donc de l'équation 4.1 :

$$\begin{aligned} \mathcal{P}_{span} &= \frac{1}{n} \sum_{p,\, \boldsymbol{x}_p \in \mathcal{S}_{SV}} H\left(\alpha_p S_p^2 - y_p f^0(\boldsymbol{x}_p) \right) \\ &= \frac{1}{n} \operatorname{Card} \left\{ \alpha_p S_p^2 > y_p f^0(\boldsymbol{x}_p \right\}. \end{aligned}$$

À noter que S_p^2 s'exprime aisément en fonction du noyau, en introduisant $\lambda_p = -1$:

$$S_p^2 = \min_{\lambda_i, i \neq p} \left\{ k \left(\sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i, \sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i \right), \quad \sum_{i=1}^{p} \lambda_i = 0 \right\}.$$

Cette borne fournit une estimation très précise de l'erreur LOO et est de loin la plus pertinente parmi celles présentées. Cependant elle présente l'inconvénient d'être discontinue par rapport aux paramètres, du fait de la présence d'une énumération. Cette contrainte est prise en compte [47] en appliquant un traitement sigmoïdal qui lisse la réponse du critère, au prix d'une complexification de ce dernier. Celui-ci est malheureusement déjà très coûteux puisqu'il implique une inversion de la matrice de Gram.

Nous n'avons donc pas exploité ce critère, malgré sa pertinence, en raison de sa trop grande complexité.

4.4 Critères basés sur la séparation de classes

Nous avons exploré durant cette thèse l'usage de critères non pas basés sur une estimation de l'erreur *Leave-One-Out* mais sur la séparabilité des classes dans l'espace transformé. Le principal critère est celui de l'Alignement du noyau que nous présentons en section 4.4.1. Nous verrons que l'expression de ce critère, construit sur une base algébrique simple, s'accompagne d'une interprétation géométrique pertinente qui fait écho au Discriminant Linéaire de Fisher, introduit dans la section 3.2. Nous présenterons par la suite d'autres critères explicitement basés sur le critère de Fisher.

4.4.1 Critère d'Alignement

On reprend ici les notations présentées dans la section 3.1 : soit un ensemble d'apprentissage $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1\ldots n}$ dont les n_1 premiers exemples appartiennent à la classe 1 ($\mathcal{S}_1 = \{(\boldsymbol{x}_i, y_i = +1)\}_{i=1,\ldots,n_1}$) et les n_2 suivants à la classe 2 ($\mathcal{S}_2 = \{(\boldsymbol{x}_i, y_i = -1)\}_{i=n_1+1,\ldots,n_n}$), et un noyau k. On pourra trouver par la suite la notation abusive n_i, où $n_i = n_1$ si $(\boldsymbol{x}_i, y_i) \in \mathcal{S}_1$ et $n_i = n_2$ si $(\boldsymbol{x}_i, y_i) \in \mathcal{S}_2$. La matrice de Gram $\boldsymbol{K}$ pour le noyau k et l'ensemble $\mathcal{S}$ est définie par $[\boldsymbol{K}]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

On définit la matrice *cible* (*target matrix*), décrivant la matrice de Gram idéale pour le problème, comme $\boldsymbol{K}^* = \boldsymbol{y}\boldsymbol{y}^T$, où $\boldsymbol{y} = [y_1, \ldots, y_n]^T$ est le vecteur des labels de classes. On peut décomposer les deux matrices introduites en blocs de classes :

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{K_{11}} & \boldsymbol{K_{12}} \\ \boldsymbol{K_{21}} & \boldsymbol{K_{22}} \end{pmatrix} \qquad \boldsymbol{K^*} = \begin{pmatrix} \mathbf{1} & \mathbf{-1} \\ \mathbf{-1} & \mathbf{1} \end{pmatrix}, \tag{4.4}$$

où $\mathbf{1}$ est une matrice dont toutes les composantes sont égales à 1. Les dimensions de cette matrice sont implicites, si bien qu'on se passera en général d'en préciser les dimensions.

Afin d'évaluer la pertinence d'un noyau pour la tâche de classification décrite par l'ensemble d'apprentissage, Cristianini et al. [56] définissent un nouveau critère basé sur une mesure de similarité exprimée par le produit scalaire de Frobenius, qui est défini entre deux matrices $\boldsymbol{A}$ et $\boldsymbol{B}$ (de termes $[\boldsymbol{A}]_{ij} = a_{ij}$ et $[\boldsymbol{B}]_{ij} = b_{ij}$) par :

$$\langle \boldsymbol{A}, \boldsymbol{B} \rangle_F = \sum_{i,j} a_{ij} b_{ij}.$$

On pourra également utiliser une notation alternative basée sur le produit de Hadamard (terme à terme) $\bullet$ et l'opérateur $\Sigma(\boldsymbol{A}) = \sum_{i,j} a_{ij}$:

$$\langle \boldsymbol{A}, \boldsymbol{B} \rangle_F = \Sigma(\boldsymbol{A} \bullet \boldsymbol{B}).$$

Les auteurs définissent ainsi le critère d'Alignement du noyau k (que nous appellerons indifféremment Alignement ou KTA, pour *Kernel Target Alignment*) comme le produit de Frobenius normalisé entre la matrice de Gram $\boldsymbol{K}$ et la matrice cible $\boldsymbol{K}^*$:

$$\mathcal{A}(\boldsymbol{K}, \boldsymbol{K^*}) = \frac{\langle \boldsymbol{K}, \boldsymbol{K^*} \rangle_F}{\|\boldsymbol{K^*}\|_F \|\boldsymbol{K}\|_F}, \tag{4.5}$$

où $\|\cdot\|_F$ est la norme associée au produit de Frobenius. On peut remarquer que

$$\|\boldsymbol{K^*}\|_F = \sqrt{\sum_{i,j=1}^{n} 1} = n.$$

La maximisation du critère d'Alignement a donc pour but d'accroître la similarité entre la matrice de Gram et la matrice cible idéale, ce qui se traduit conjointement par l'accroissement conjoint de la similarité (mesurée par la fonction noyau) entre les exemples de même classe, et sa réduction entre les exemples de classes opposées. La normalisation du produit de Frobenius permet de soustraire au critère l'influence d'un facteur d'échelle, et restreint ainsi le critère d'Alignement à un intervalle standard :

$$-1 \leq \mathcal{A}(\boldsymbol{K}, \boldsymbol{K}^*) \leq 1. \tag{4.6}$$

Dans le cas de classes mal proportionnées ($n_1 \gg n_2$ ou inversement) on peut compenser la représentation des classes en utilisant la matrice cible *pondérée* $\hat{\boldsymbol{K}}^* = \hat{\boldsymbol{y}}\hat{\boldsymbol{y}}^{\boldsymbol{T}}$ où $\hat{y}_i = \frac{y_i}{n_i}$. On a alors :

$$\hat{\boldsymbol{K}}^* = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{y_i}{n_i}\frac{y_j}{n_j} k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{4.7}$$

$$= \begin{pmatrix} \frac{1}{n_1^2}\mathbf{1} & -\frac{n_1}{n_2}\mathbf{1} \\ -\frac{n_1}{n_2}\mathbf{1} & \frac{1}{n_2^2}\mathbf{1} \end{pmatrix} = \frac{1}{n_1 n_2} \begin{pmatrix} \frac{n_2}{n_1}\mathbf{1} & -\mathbf{1} \\ -\mathbf{1} & \frac{n_1}{n_2}\mathbf{1} \end{pmatrix} \tag{4.8}$$

$$\text{et} \quad ||\hat{\boldsymbol{K}}^*||_F = \frac{n}{n_1 n_2}. \tag{4.9}$$

4.4.1.1 Interprétation géométrique

En développant le produit de Frobenius entre la matrice de Gram et la matrice cible pondérée, en terme de produits scalaires dans l'espace transformé, il est possible de faire ressortir une

interprétation géométrique du critère d'Alignement :

$$\begin{aligned}
\left\langle \boldsymbol{K}, \hat{\boldsymbol{K}}^* \right\rangle_F &= \left\langle \begin{pmatrix} \boldsymbol{K_{11}} & \boldsymbol{K_{12}} \\ \boldsymbol{K_{21}} & \boldsymbol{K_{22}} \end{pmatrix}, \begin{pmatrix} \frac{1}{n_1^2}\mathbf{1} & -\frac{n_1}{n_2}\mathbf{1} \\ -\frac{n_1}{n_2}\mathbf{1} & \frac{1}{n_2^2}\mathbf{1} \end{pmatrix} \right\rangle_F \\
&= \frac{1}{n_1^2} \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{S}_1} k(\boldsymbol{x_i}, \boldsymbol{x_j}) + \frac{1}{n_2^2} \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{S}_2} k(\boldsymbol{x_i}, \boldsymbol{x_j}) - \frac{2}{n_1 n_2} \sum_{\boldsymbol{x}_i \in \mathcal{S}_1} \sum_{\boldsymbol{x}_j \in \mathcal{S}_2} k(\boldsymbol{x_i}, \boldsymbol{x_j}) \\
&= \frac{1}{n_1^2} \Big(\sum_{\boldsymbol{x}_i \in \mathcal{S}_1} \phi(\boldsymbol{x}_i) \Big)^2 + \frac{1}{n_2^2} \Big(\sum_{\boldsymbol{x}_j \in \mathcal{S}_2} \phi(\boldsymbol{x}_j) \Big)^2 - \frac{2}{n_1 n_2} \sum_{\boldsymbol{x}_i \in \mathcal{S}_1} \phi(\boldsymbol{x}_i) \sum_{\boldsymbol{x}_j \in \mathcal{S}_2} \phi(\boldsymbol{x}_j) \\
&= \left(\frac{1}{n_1} \sum_{\boldsymbol{x}_i \in \mathcal{S}_1} \phi(\boldsymbol{x}_i) - \frac{1}{n_2} \sum_{\boldsymbol{x}_j \in \mathcal{S}_2} \phi(\boldsymbol{x}_j) \right)^2 \\
&= \left\| \boldsymbol{\mu}_1^\Phi - \boldsymbol{\mu}_2^\Phi \right\|^2,
\end{aligned}$$

où $\boldsymbol{\mu}_c^\Phi = \frac{1}{n_c} \sum_{\boldsymbol{x}_i \in \mathcal{S}_c} \phi(\boldsymbol{x}_i)$ est le centre des exemples de la classe c dans l'espace transformé.

La maximisation de l'Alignement se traduit donc dans l'espace transformé par la maximisation de la distance dite *inter-classes* entre les centres des deux classes, approche suivie par [251] sans référence au KTA. On retrouve ainsi le numérateur du critère de Fisher (équation 3.3) exprimant le vecteur normal de l'hyperplan de séparation optimale. Néanmoins, tandis que le critère de Fisher fait intervenir dans son dénominateur les covariances des exemples des classes, le dénominateur du critère d'Alignement est difficilement interprétable géométriquement puisqu'il fait intervenir des produits scalaires au carré dans l'espace transformé :

$$||\boldsymbol{K}||_F = \sqrt{\sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{S}} \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle^2}.$$

Ceci étant, on peut montrer [56] que la mesure d'Alignement est proportionnelle à la distance inter-classes et inversement proportionnelle aux distances intra-classes.

4.4.1.2 Fenêtres de Parzen

Nous apportons ici une interprétation nouvelle du critère d'Alignement en mettant en lumière une relation particulière du noyau gaussien RBF avec les fenêtres de Parzen. En 1962, Parzen propose [177] une nouvelle méthode d'estimation de la densité de probabilité basée sur ce qu'il appelle les fonctions noyaux, qui n'ont à priori pas de lien direct avec les noyaux impliqués dans les SVM. Nous emploierons de préférence le terme équivalent de « fenêtres de Parzen », pour éviter toute ambiguité.

Le problème posé est celui de l'estimation d'une densité de probabilité $f(\boldsymbol{x})$ à partir de n échantillons i.i.d. $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ d'une variable aléatoire $\boldsymbol{x}$. Une méthode courante est l'estimation par histogramme mais celle-ci présente l'inconvénient de fournir une réponse fortement discontinue et de support restreint.

On appelle *fenêtres de Parzen* toute fonction $K(y)$ intégrable à valeurs réelles non-négatives, si elle remplit les conditions suivantes :

- K est de somme unitaire, $\int_{-\infty}^{\infty} K(y)dy = 1$
- K est symétrique, $K(-y) = K(y)$

L'estimation de densité consiste simplement à sommer les contributions de la fenêtre au voisinage des exemples de l'échantillon, soit :

$$\hat{f}_n(\boldsymbol{x}) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right).$$

Cette méthode intuitive, statistiquement justifiée par Parzen, permet ainsi de lisser la contribution de chaque exemple sur la densité de probabilité estimée. En pratique on choisit en général des fenêtres dont le maximum est atteint en 0, et décroissants au voisinage, comme les exemples suivants :

- Fenêtre uniforme $K_U(y) = \frac{1}{2}\mathbf{1}_{\{|y|\leq 1\}}$
- Fenêtre triangulaire $K_{tri}(y) = (1-|y|)\mathbf{1}_{\{|y|\leq 1\}}$
- Fenêtre cosinus $K_{cos}(y) = \frac{\pi}{4}\cos\left(\frac{\pi}{2}y\right)\mathbf{1}_{\{|y|\leq 1\}}$
- Fenêtre gaussienne $K_G(y) = (2\pi)^{-\frac{1}{2}}e^{-\frac{1}{2}y^2}$

On remarque que la fenêtre uniforme équivaut à une estimation par histogramme, en relâchant la contrainte de pas fixe entre les *bins*. Parmi les fenêtres présentées, la fenêtre gaussienne est particulièrement intéressante puisqu'elle est continue et infiniment dérivable en tout point. De plus, on remarque que le lien avec le noyau gaussien RBF présenté précédemment est immédiat puisque

$$k_{\text{rbf}}(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}\right) = \sqrt{2\pi}\,K_G\left(\frac{\|\boldsymbol{x}-\boldsymbol{y}\|}{\sigma}\right).$$

Dans le cas d'un problème à deux classes, on peut ainsi caractériser, à l'aide de l'estimation de Parzen sur la fenêtre gaussienne, la densité de probabilité pour chacune des classes :

$$\begin{aligned}\hat{f}(\boldsymbol{x}|y=+1) &= \frac{1}{n_1\sigma}\sum_{\boldsymbol{x}_i\in\mathcal{S}_1} K_G\left(\frac{\|\boldsymbol{x}_i-\boldsymbol{x}\|}{\sigma}\right)\\ \hat{f}(\boldsymbol{x}|y=-1) &= \frac{1}{n_2\sigma}\sum_{\boldsymbol{x}_i\in\mathcal{S}_2} K_G\left(\frac{\|\boldsymbol{x}_i-\boldsymbol{x}\|}{\sigma}\right).\end{aligned}$$

Un problème de discrimination est d'autant plus facile à résoudre qu'il est séparable. Nous définissons ainsi un critère de séparabilité $\mathcal{P}_{sep}$ sur chaque exemple en soustrayant les densités de probabilité des deux classes :

$$\begin{aligned}\mathcal{P}_{sep}(\boldsymbol{x}_j,y_j) &= \hat{f}(\boldsymbol{x}_j|y=y_j) - \hat{f}(\boldsymbol{x}_j|y\neq y_j)\\ &= y_j\sum_{y_i\in\{-1,+1\}} y_i\,\hat{f}(\boldsymbol{x}_j|y=y_i).\end{aligned}$$

Ce critère peut être interprété comme un détecteur d'*outlier*. En effet, il mesure l'adéquation de la classe y associée à l'exemple $\boldsymbol{x}$ par rapport aux densités de probabilité des deux classes dans son voisinage. On remarque que $\mathcal{P}_{sep}$ évolue dans l'intervalle $[-1;+1]$ et est d'autant plus proche de 1 que la probabilité que l'exemple $\boldsymbol{x}_i$ soit de classe y_i est élevée. Ainsi, si l'on somme le critère $\mathcal{P}_{sep}$ sur tous les exemples de l'ensemble d'apprentissage en le pondérant par le nombre d'exemples de chaque classe, on définit un critère d'estimation de séparabilité de l'ensemble $\mathcal{S}$:

$$\begin{aligned}\mathcal{P}_{sep}(\mathcal{S}) &= \frac{1}{n_1}\sum_{(\boldsymbol{x}_j,y_j)\in\mathcal{S}_1}^{n}\mathcal{P}_{sep}(\boldsymbol{x}_j,y_j) + \frac{1}{n_2}\sum_{(\boldsymbol{x}_j,y_j)\in\mathcal{S}_2}^{n}\mathcal{P}_{sep}(\boldsymbol{x}_j,y_j)\\ &= \sum_{j=1}^{n}\frac{y_j}{n_j}\sum_{y_i\in\{-1,+1\}} y_i\,\hat{f}(\boldsymbol{x}_j,y=y_i)\\ &= \sum_{j=1}^{n}\frac{y_j}{n_j}\left[y_1\frac{1}{n_1\sigma}\sum_{\boldsymbol{x}_i\in\mathcal{S}_1}K_G\left(\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|}{\sigma}\right) + y_2\frac{1}{n_2\sigma}\sum_{\boldsymbol{x}_i\in\mathcal{S}_2}K_G\left(\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|}{\sigma}\right)\right]\\ &= \frac{1}{\sigma}\sum_{j=1}^{n}\sum_{i=1}^{n}\frac{y_j}{n_j}\frac{y_i}{n_i}K_G\left(\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|}{\sigma}\right).\end{aligned}$$

D'où, d'après les relations 4.10 et 4.7 :

$$\mathcal{P}_{sep}(\mathcal{S}) = \frac{1}{\sqrt{2\pi}\sigma}\left\langle \boldsymbol{K},\hat{\boldsymbol{K}}^*\right\rangle_F,$$

où $\boldsymbol{K}$ est la matrice de Gram du noyau k_{rbf} de paramètre σ, et $\hat{\boldsymbol{K}}^*$ la matrice cible de l'ensemble $\mathcal{S}$.

Nous avons donc montré que dans le cas de noyaux (au sens des SVM) respectant les conditions de Parzen, comme c'est le cas pour le noyau gaussien RBF, le produit de Frobenius, qui constitue le critère non normalisé du KTA, peut être interprété comme une mesure de séparabilité de l'ensemble d'apprentissage, liée à l'estimation de Parzen de la densité de probabilité des deux classes.

4.4.1.3 Dérivation

De par l'expression très simple du produit de Frobenius, la dérivation du critère d'Alignement est immédiate. Si l'on considère un noyau $k_{\boldsymbol{\Theta}}$ caractérisé par les paramètres $\boldsymbol{\Theta} = [\theta_1, \ldots, \theta_p]$, et la matrice de Gram correspondante $\boldsymbol{K}_{\boldsymbol{\Theta}}$, alors :

$$\frac{\partial}{\partial \theta_p} \langle \boldsymbol{K}_{\boldsymbol{\Theta}}, \boldsymbol{K}^* \rangle_F = \left\langle \partial_{\theta_p} \boldsymbol{K}_{\boldsymbol{\Theta}}, \boldsymbol{K}^* \right\rangle_F \tag{4.10}$$

$$\frac{\partial}{\partial \theta_p} ||\boldsymbol{K}_{\boldsymbol{\Theta}}||_F = \frac{\left\langle \partial_{\theta_p} \boldsymbol{K}_{\boldsymbol{\Theta}}, \boldsymbol{K}_{\boldsymbol{\Theta}} \right\rangle_F}{||\boldsymbol{K}_{\boldsymbol{\Theta}}||_F}, \tag{4.11}$$

où l'on a défini les matrices $\partial_{\theta_p} \boldsymbol{K}_{\boldsymbol{\Theta}} = [\partial_{\theta_p} k_{\boldsymbol{\Theta}}(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{ij}$. Ainsi en calculant ces matrices on peut dériver l'Alignement par rapport à $\boldsymbol{\Theta}$:

$$\frac{\partial}{\partial \theta_p} \mathcal{A}(\boldsymbol{K}_{\boldsymbol{\Theta}}, \boldsymbol{K}^*) = \frac{\left\langle \partial_{\theta_p} \boldsymbol{K}_{\boldsymbol{\Theta}}, \boldsymbol{K}^* \right\rangle_F}{||\boldsymbol{K}_{\boldsymbol{\Theta}}||_F ||\boldsymbol{K}^*||_F} - \frac{\langle \boldsymbol{K}_{\boldsymbol{\Theta}}, \boldsymbol{K}^* \rangle_F \left\langle \boldsymbol{K}_{\boldsymbol{\Theta}}, \partial_{\theta_p} \boldsymbol{K}_{\boldsymbol{\Theta}} \right\rangle_F}{||\boldsymbol{K}_{\boldsymbol{\Theta}}||_F^3 ||\boldsymbol{K}^*||_F}. \tag{4.12}$$

On peut ainsi appliquer les techniques d'optimisation présentées précédemment sur le critère d'Alignement pour affiner les paramètres du noyau $k_{\boldsymbol{\Theta}}$.

4.4.1.4 Critique de l'Alignement

Nous avons expliqué que la normalisation du produit de Frobenius restreint l'Alignement à l'intervalle $[-1; +1]$ (équation 4.6). Pourtant si l'on développe l'expression de la matrice cible,

$$\boldsymbol{K}^* = \boldsymbol{y}\boldsymbol{y}^T,$$

on en déduit la positivité du produit de Frobenius

$$\langle \boldsymbol{K}, \boldsymbol{K}^* \rangle_F = \boldsymbol{y}\boldsymbol{K}\boldsymbol{y}^T \geq 0,$$

puisque comme nous l'avons vu, tout noyau respectant la condition de Mercer a sa matrice de Gram semi-définie positive sur tout ensemble d'exemples. Les termes de normalisation de l'Alignement étant positifs, on en déduit

$$0 \leq \mathcal{A}(\boldsymbol{K}, \boldsymbol{K}^*) \leq 1.$$

On peut remarquer en outre que certains noyaux, comme le noyau gaussien RBF, ont toujours une valeur positive, ce qui implique donc que les exemples sont restreints, dans l'espace transformé, à un cône d'angle borné par $\frac{\pi}{2}$ puisque tous les produits scalaires y sont positifs. Ainsi, dans le meilleur des cas, les exemples de classes opposées sont orthogonaux ($k(\boldsymbol{x}_+, \boldsymbol{x}_-) = 0$). La matrice de Gram optimale a donc pour valeur :

$$\boldsymbol{K}_{opt} = \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}.$$

L'Alignement est donc borné par la valeur suivante :

$$\mathcal{A}(\boldsymbol{K}_{opt}, \boldsymbol{K}^*) \leq \frac{\sqrt{n_1^2 + n_2^2}}{n}.$$

Soit, dans le cas de classes également réparties,

$$\mathcal{A}(\boldsymbol{K}_{opt}, \boldsymbol{K}^*) \leq \frac{1}{\sqrt{2}}.$$

On remarque donc que l'intervalle dans lequel évolue la mesure d'Alignement est fortement déterminé par le choix du noyau, ce qui remet en cause sa fiabilité, par exemple pour comparer la pertinence de deux noyaux différents. Cette carence s'explique très simplement en termes géométriques. Dans le cas du noyau gaussien, les exemples sont situés dans un cône dont l'extrémité est à l'origine de l'espace ; ils sont par ailleurs situés sur l'intersection de ce cône avec la sphère unité puisque

$$k_{\text{rbf}}(\boldsymbol{x}, \boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}\|^2}{2\sigma^2}\right) = 1.$$

L'origine est donc excentrée par rapport au centre de l'ensemble des exemples. Or la définition de la matrice cible suppose que, dans l'idéal, les exemples de classes opposés sont situés de part et d'autre de l'origine. En d'autres termes, la maximisation du critère d'Alignement est une condition suffisante à l'optimisation des performances, mais non nécessaire.

C'est là la critique principale au critère d'Alignement que l'on retrouve dans la littérature [167][188][189] et qui peut d'ailleurs être adressée également aux Machines à Vecteurs de Support [152]. Une solution à ce problème consiste à translater les exemples dans l'espace transformé pour les centrer autour de l'origine, comme le font par exemple Meila [152] ou Pothin et Richard [189]. Mais ce genre de solution implique généralement un procédure d'optimisation portant sur autant de coefficients que d'exemples dans la base. La procédure devient donc trop coûteuse dans le cas de bases d'apprentissage conséquentes (plusieurs milliers d'exemples).

4.4.2 Séparabilité dans l'espace Transformé (KCS)

Construit sur des bases algébriques, le produit de Frobenius présent dans le critère d'Alignement est en fait, comme nous l'avons vu, égal à la mesure de distance inter-classes dans l'espace transformé que l'on retrouve au numérateur du critère de Fisher. Il est en fait possible d'exprimer pleinement le critère de Fisher dans l'espace transformé. On le retrouve également dans la littérature, désigné par les termes de « critère de séparabilité des classes », défini de la façon suivante :

$$J = \frac{\operatorname{tr} \boldsymbol{S}_b}{\operatorname{tr} \boldsymbol{S}_w}, \tag{4.13}$$

où $\boldsymbol{S_b}$ est la matrice de *dispersion inter-classes* (b pour *between-class scatter*) et $\boldsymbol{S_w}$ la matrice de *dispersion intra-classe* (w pour *within-class scatter*). Ces dernières ont les expressions suivantes :

$$\boldsymbol{S}_b = \frac{1}{n} \sum_{c=1,2} n_c (\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^T \tag{4.14}$$

$$\boldsymbol{S}_w = \sum_{c=1,2} \sum_{\boldsymbol{x}_i \in \mathcal{S}_c} (\boldsymbol{x}_i - \boldsymbol{\mu}_c)(\boldsymbol{x}_i - \boldsymbol{\mu}_c)^T, \tag{4.15}$$

où $\boldsymbol{\mu}_c = \frac{1}{n_c} \sum_{\boldsymbol{x}_i \in \mathcal{S}_c} \boldsymbol{x}_i$ est le centre des exemples de la classe c et $\boldsymbol{\mu}$ le centre de l'ensemble des exemples ($\boldsymbol{\mu} = \frac{1}{n} \sum_{\boldsymbol{x}_i \in \mathcal{S}} \boldsymbol{x}_i = \frac{1}{n} (n_1 \boldsymbol{\mu}_1 + n_2 \boldsymbol{\mu}_2)$). Dans le cas de classes également réparties ($\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \frac{\boldsymbol{\mu}}{2}$), on retrouve dans le quotient des deux matrices de dispersion une expression très proche du critère de Fisher énoncé en section 3.2 :

$$\boldsymbol{w}_F = \frac{\boldsymbol{S}_b}{\boldsymbol{S}_w} = \frac{1}{4} \frac{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^2}{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2},$$

où l'on suppose que la matrice de dispersion intra-classe $\boldsymbol{S_w}$ est inversible.

Les matrices $\boldsymbol{S_b}$ et $\boldsymbol{S_w}$ pouvant s'exprimer exclusivement en terme de produits scalaires sur les exemples, il est possible [252][240] d'y substituer l'usage d'une fonction noyau pour « kerneliser » l'expression du critère J (équation 4.13) :

$$\mathcal{J} = \frac{\mathbf{1}_n^T \boldsymbol{B} \mathbf{1}_n}{\mathbf{1}_n^T \boldsymbol{W} \mathbf{1}_n} = \frac{\Sigma(\boldsymbol{B})}{\Sigma(\boldsymbol{W})}, \tag{4.16}$$

où l'on rappelle que l'opérateur Σ correspond à la somme de tous les termes d'une matrice. Les matrices kernelisées de dispersion inter-classes ($\boldsymbol{B}$) et intra-classes ($\boldsymbol{W}$), introduites dans la relation précédente, ont les expressions suivantes :

$$\boldsymbol{B} = \begin{pmatrix} \frac{1}{n_1}\boldsymbol{K}_{11} & \mathbf{0} \\ \mathbf{0} & \frac{1}{n_2}\boldsymbol{K}_{22} \end{pmatrix} - \boldsymbol{K} \tag{4.17}$$

$$\boldsymbol{W} = \begin{pmatrix} k_{11} & 0 & \cdots & 0 \\ 0 & k_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & k_{nn} \end{pmatrix} - \begin{pmatrix} \frac{1}{n_1}\boldsymbol{K}_{11} & \mathbf{0} \\ \mathbf{0} & \frac{1}{n_2}\boldsymbol{K}_{22} \end{pmatrix}. \tag{4.18}$$

On utilise ici la décomposition de la matrice de Gram par blocs de classes introduite dans l'équation 4.4. Le critère $\mathcal{J}$ introduit est donc une mesure de séparabilité dans l'espace transformé que nous désignerons par l'acronyme KCS (pour *Kernel Class Separability*).

Dérivation et régularisation

Le critère $\mathcal{J}$ est également aisément dérivable et sa dérivée par rapport au paramètre θ_p implique la matrice $\partial_{\theta_p}\boldsymbol{K}_{\boldsymbol{\Theta}}$ introduite précédemment.

Toutefois, dans le cas particulier du noyau RBF gaussien, l'expression du critère KCS implique une instabilité numérique dans la maximisation de $\mathcal{J}$ en provoquant systématiquement la convergence des deux dispersions vers 0 (soit $\operatorname{tr} S_b \xrightarrow[\sigma\to\infty]{} 0$ et $\operatorname{tr} S_w \xrightarrow[\sigma\to\infty]{} 0$, dont le rapport tend vers 1, borne supérieure du critère). On contourne ce problème en appliquant une régularisation sur le dénominateur :

$$\tilde{\mathcal{J}} = \frac{\mathbf{1}_n^T \boldsymbol{B} \mathbf{1}_n}{\mathbf{1}_n^T \boldsymbol{W} \mathbf{1}_n} = \frac{\Sigma(\boldsymbol{B})}{\Sigma(\boldsymbol{W}) + \epsilon}, \tag{4.19}$$

qui évite la convergence $\mathcal{J} \xrightarrow[\sigma\to\infty]{} 1$ sans avoir à faire appel aux techniques d'optimisation sous contraintes, qui complexifieraient l'algorithme. Nous verrons par la suite que cette procédure de régularisation est également nécessaire lorsque nous introduirons le critère KCS pour la sélection automatique de descripteurs (voir section 7.5.4).

4.5 Facteur d'erreur C

Le facteur d'erreur C est un paramètre particulier puisqu'il est le seul à ne pas intervenir dans la définition du noyau. Nous avons vu dans la section 3.6, que celui-ci intervient différemment dans les Machines à Marge Souple selon le choix de la norme appliquée sur les variables d'écart. On peut cependant tenter d'en donner une interprétation intuitive :

- Lorsque $C \to \infty$, la tolérance aux erreurs de classification est de plus en plus rigide. On voit que pour les deux problèmes $L1$ et $L2$ (équations 3.30 et 3.29), on retrouve alors l'expression du problème à Marge Dure. Le problème est en effet équivalent dans le cas de données séparables puisque les variables d'écart sont nulles.
- Lorsque $C \to 0$, le système tolère les erreurs jusqu'à ne plus distinguer les exemples des deux classes. Le cas extrême $C = 0$ implique d'ailleurs $\alpha_i = 0\ \forall i$ pour les $L1$ SVM, ce qui signifie que la fonction de décision ne dépend plus des exemples d'apprentissage. De même pour les $L2$ SVM où la matrice $\boldsymbol{H}$ devient négligeable dans l'expression du problème dual (équation 3.29).

La valeur optimale de C constituera donc un compromis entre la tolérance aux *outliers* et la minimisation de l'erreur.

Si l'on utilise la norme $L2$, l'expression du problème est la même que dans le cas des Machines à Marge Dure, en ajoutant la constante $\frac{1}{C}$ aux termes diagonaux de la matrice de Gram. La plupart des méthodes d'optimisation des paramètres étant basées sur la matrice de Gram, il

est donc possible d'assimiler la constante C à un hyper-paramètre du noyau dans le cas des $L2$ SVM.

Toutefois, concernant la norme $L1$, plus généralement employée, le raisonnement précédent ne s'applique pas. Le contrôle du risque structurel est d'ailleurs moins simple dans le cadre des Machines à Marge Souple car les résultats de la théorie de Vapnik-Chervonenkis ne s'appliquent pas tels quels. Néanmoins, Steinwart a montré [221] que pour tout $\epsilon > 0$, il existe une valeur C_ϵ telle que pour tout $C \geq C_\epsilon$, le risque de la fonction de décision obtenue par $L1$ SVM n'excède pas de plus de ϵ le risque fonctionnel minimal.

Il existe peu de travaux dans la littérature sur la détermination automatique de la valeur optimale de C. Nous apportons tout de même ici une réponse dans le cas des $L1$ SVM.

4.5.1 Valeur de Joachims

Dans son logiciel *SVMlight* [113], Joachims propose la valeur par défaut suivante :

$$C_{def} = \frac{1}{\bar{R}^2}, \tag{4.20}$$

où $\bar{R}$ est la distance moyenne des exemples à l'origine dans l'espace transformé. Cette valeur est similaire au rayon R introduit précédemment, qui concerne la sphère minimale contenant tous les exemples dans l'espace transformé. Plusieurs méthodes existent pour évaluer ces grandeurs, qui sont présentées dans l'annexe A. Celle qu'emploie Joachims est présentée dans la section A.3.

Validation expérimentale

Nous montrons par une courte expérience qu'en pratique la valeur de Joachims constitue le meilleur compromis entre complexité et performance. Afin de valider celle-ci nous apprenons une machine SVM à noyau gaussien RBF (où le paramètre σ est fixé arbitrairement à 1) pour chaque valeur C d'un ensemble de 25 valeurs réparties logarithmiquement entre 10^{-6} et 10^6. Pour chaque machine nous évaluons l'erreur d'apprentissage, qui du fait du sur-apprentissage, se révèle fortement biaisée, et l'erreur *Leave-one-out*, qui constitue, comme nous l'avons expliqué, l'estimateur le plus fiable de l'erreur de généralisation. L'évolution du taux de vecteurs de support parmi les exemples d'apprentissage, ainsi que le temps d'apprentissage nous apporterons également des informations utiles quant à la complexité de la phase d'apprentissage.

Nous avons appliqué cette évaluation sur trois problèmes de discrimination. Chacun d'entre eux se résume à une base de données contenant les exemples des deux classes, caractérisés par une série des descripteurs propres aux problèmes. Nous avons ainsi exploité deux bases publiques disponibles sur le dépôt UCI [16], destiné à offrir des données communes à la communauté scientifique en apprentissage statistique : *Spambase* qui décrit un problème de détection de mails parasites, et *Ionosphere*, qui concerne la détection d'une structure dans l'ionosphère. Une troisième base a été constituée sur les données du corpus ESTER (présenté dans la partie expérimentale finale, section 10.1.1) et décrit un problème de discrimination entre parole et musique pures, caractérisé par une grande collection de descripteurs. On trouvera plus de détail sur ces bases expérimentales dans l'annexe B. Celles-ci seront par ailleurs exploitées à nouveau dans la section suivante (4.6), sur les critères de sélection, ainsi que dans le chapitre 7 traitant des méthodes de sélection de descripteurs.

Les figures 4.5, 4.6 et 4.7 montrent les résultats de l'expérience sur les bases respectives *Spambase*, *Ionosphere* et *parole/musique*. On constate en premier lieu que les grandeurs représentées ont un profil commun dans les trois cas. Ainsi lorsque $\log_{10}(C)$ est inférieur à une certaine valeur (autour de -2), le taux d'erreur d'apprentissage et l'erreur *Leave-one-out* sont égaux et constants, de même que le taux de vecteurs de support, très élevé (de l'ordre de 80%). Ceci s'explique par le fait que la pénalisation des erreurs (des variables d'écart) est négligeable devant la minimisation de la marge, dont il résulte un hyperplan de séparation quelconque, et une large proportion d'exemples mal classifiés (et donc de vecteurs de support au delà de la marge). On constate par ailleurs que le temps d'apprentissage décroît très vite lorsque C augmente.

Passé le seuil autour de -2, on constate une rapide décroissance des deux erreurs, provenant du fait que les variables d'écart participent à l'optimisation. La réduction du nombre d'exemples mal

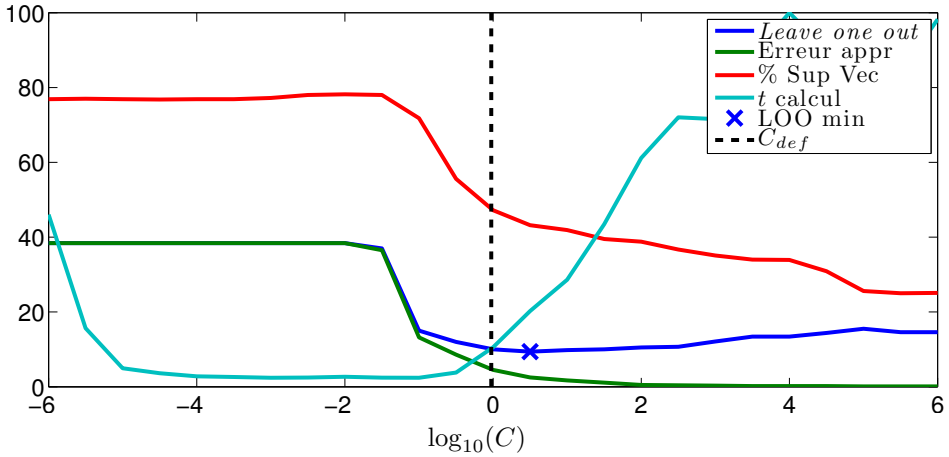


FIGURE 4.5 – Évolution des erreurs *Leave-one-out* et d'apprentissage, ainsi que du taux de vecteurs de support et du temps d'apprentissage (ici en pourcentage par rapport au temps maximal observé), par rapport aux variations du facteur C, sur la base *Spambase*. La ligne noire pointillée indique la valeur C_{def} définie par Joachims, et la croix bleue le minimum de l'erreur *Leave-one-out*. Le temps de calcul relatif (*t* calcul) figure ici en pointillés à titre informatif.

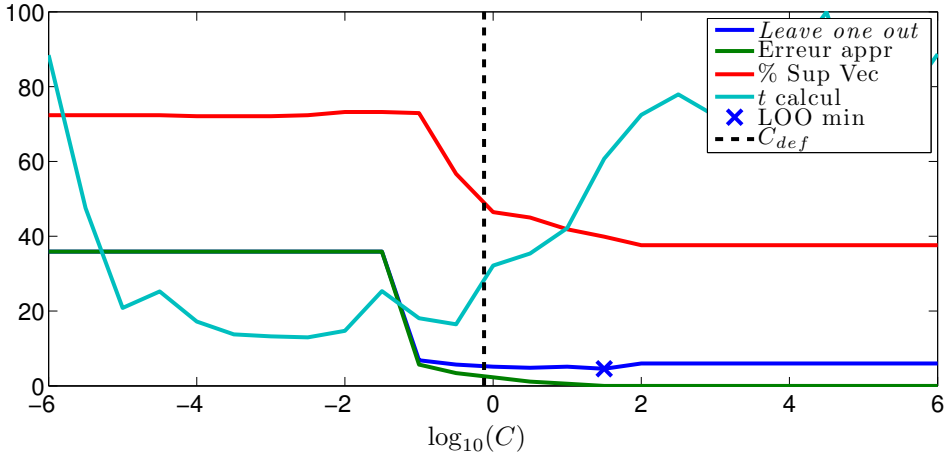


FIGURE 4.6 – Résultats de la même expérience sur la base *Ionosphere*.

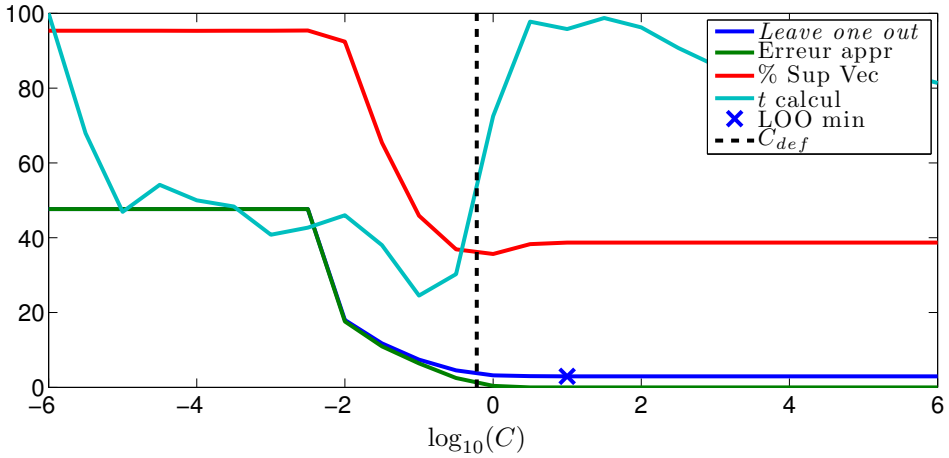


FIGURE 4.7 – Résultats de la même expérience sur la base *parole/musique*.

classifiés implique nécessairement la réduction du nombre de vecteurs de support (puisque, nous le rappelons, tout exemple mal classifié est un vecteur de support auquel est associé une variable d'écart non nulle). Cependant, plus on augmente le facteur C, plus le compromis entre marge et variables d'écart est complexe, et plus la convergence de l'algorithme d'apprentissage est longue, d'où un temps de calcul qui augmente de manière quasi monotone.

Tandis que l'erreur d'apprentissage décroît de manière monotone avec l'augmentation de C, on constate que l'erreur *Leave-one-out* augmente de nouveau après avoir dépassé un minimum global (marqué d'une croix bleue sur les figures), ce qui traduit le phénomène de sur-apprentissage des exemples de la base, que l'erreur d'apprentissage ne permet pas de constater.

La ligne verticale pointillée représente la valeur C_{def} calculée sur la base des exemples. On constate dans les trois cas que celle-ci est proche du minimum de l'erreur LOO, ou au moins, que la valeur de l'erreur LOO y est très proche de sa valeur minimale, ce qui confirme la pertinence de cette valeur en termes de performances. Enfin, on remarque dans les trois cas également, que le léger incrément d'erreur apporté par C_{def} est compensé par une nette réduction en temps de calcul par rapport à la valeur minimisant l'erreur LOO.

La valeur de Joachims constitue donc un excellent compromis entre complexité et performances, et se révèle très simple et rapide à calculer.

4.5.2 Inclusion du facteur C dans les critères

L'expérience précédente nous montre l'influence déterminante du facteur C sur les performances des Machines à Vecteurs de Support. Pourtant les critères de séparabilité introduits dans la section 4.4 (alignement et séparabilité de classes) sont exclusivement basés sur la matrice de Gram, qui synthétise l'action du noyau sur les exemples d'apprentissage. Ils n'incluent donc pas le facteur d'erreur C, qui est un paramètre extérieur au noyau.

En introduisant les machines à marge souple dans ce document (section 3.6) nous avons mentionné les deux principaux paradigmes $L1$ et $L2$ qui diffèrent selon que l'on applique la puissance $k = 1$ ou $k = 2$ à la somme des variables d'écart, pour constituer la pénalité totale des exemples mal classifiés, pondérée par le facteur C.

Le cas $L1$ est le plus généralement employé parce qu'il permet de conserver le même problème d'optimisation que dans le cas des machines à marge dure, en ajoutant seulement une borne supérieure sur les facteurs de Lagrange (voir équation 3.30). On peut montrer cependant que le cas $L2$ est équivalent au seul ajout de la constante $\frac{1}{C}$ sur les termes diagonaux de la matrice de Gram (équation 3.29). On définit donc la matrice de Gram *ajustée* par le facteur C comme suit :

$$\boldsymbol{K}_C = \boldsymbol{K} + \frac{1}{C}\boldsymbol{I}$$

Bien que nous employions exclusivement le paradigme $L1$ dans l'implémentation des SVM, nous proposons d'appliquer le principe de l'approche $L2$ sur les matrices de Gram exploitées pour le calcul de l'Alignement et du critère KCS. Les deux problèmes à marge souple ne sont pas équivalents formellement, de sorte que les paramètres n'ont théoriquement pas la même influence. Toutefois, nous verrons dans la section expérimentale 4.6 qu'en pratique cette opération renforce la fiabilité des critères sus-cités.

En suivant le raisonnement précédent, on en déduit que la matrice de Gram non-ajustée est équivalente à un problème où $C = \infty$, soit une pénalisation infinie des exemples hors de la marge, ce qui revient à appliquer un modèle à marge dure sur un problème non-séparable. En pratique on trouve tout de même une solution dans ce cas, mais celle-ci est sous-optimale.

Cet apport se révèle par ailleurs d'un coût négligeable puisque son application revient à ajouter

un terme constant aux grandeurs impliquées dans les critères :

$$\begin{aligned}
\langle \boldsymbol{K}_C, \boldsymbol{K}^* \rangle_F &= \langle \boldsymbol{K}, \boldsymbol{K}^* \rangle_F + \frac{n}{C} \\
\|\boldsymbol{K}_C\|_F^2 &= \|\boldsymbol{K}\|_F^2 + \frac{n}{C^2} \\
\Sigma(\boldsymbol{B}_C) &= \Sigma(\boldsymbol{B}) - \frac{n-2}{C} \\
\Sigma(\boldsymbol{W}_C) &= \Sigma(\boldsymbol{W}) + \frac{n-2}{C}
\end{aligned}$$

où $\boldsymbol{B}_C$ et $\boldsymbol{W}_C$ sont le résultat de l'application des formules 4.17 et 4.18 sur la matrice de Gram ajustée ; $\Sigma(\boldsymbol{B})$ et $\Sigma(\boldsymbol{W})$ interviennent dans la définition du critère KCS (équation 4.16).

4.6 Evaluation des critères de sélection de noyau

L'expérience pratique décrite dans cette section confirme les remarques avancées sur les avantages et les limites des critères proposés pour l'évaluation du noyau. Nous étudions pour cela la recherche du paramètre $\hat{\sigma}$ optimal sur un noyau RBF gaussien. A titre expérimental, on déploie donc une recherche par maillage (dont la complexité reste ici raisonnable sur une dimension) en faisant varier le paramètre σ sur 25 valeurs réparties logarithmiquement entre 0.1 et 20, qui constituent des bornes assez larges pour ce paramètre. On rappelle que l'introduction du facteur $\frac{1}{d}$ dans l'expression des noyaux (voir section 3.5.3) réduit largement le champ des valeurs optimales mesurées pour σ, qui se trouvent généralement dans le voisinage de 1.

La valeur optimale retenue est celle minimisant le critère, dans le cas des critères d'estimation de l'erreur *Leave-one-out*, décrits dans la section 4.3, ou le maximisant, dans le cas des critères de séparabilité de classes décrits dans la section 4.4.

On a également calculé l'erreur *Leave-one-out* exacte pour chacune des valeurs de σ, qui nous sert de référence pour estimer l'erreur de généralisation.

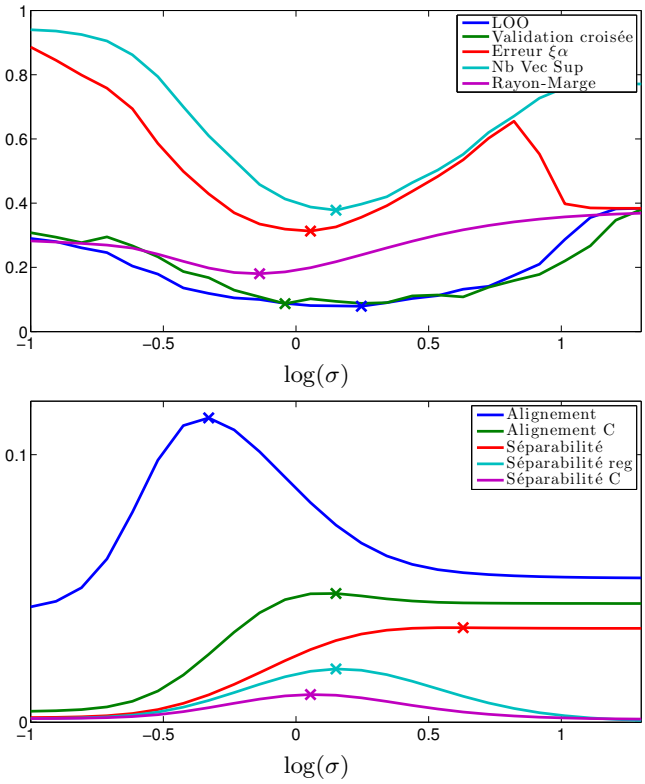
Nous avons appliqué cette évaluation sur les bases introduites dans la section 4.5.1, auxquelles nous ajoutons la base *Lymphoma*. Ces bases, ainsi que les problèmes qu'elles modélisent, sont décrits en détail dans l'annexe B.

Dans un premier temps nous avons fixé le facteur d'erreur à $C = 1$ dans toutes les étapes (apprentissage des SVM, et ajustement éventuel des matrices de Gram, d'après la procédure proposée dans la section 4.5.2). Les figures 4.8, 4.9 et 4.10 illustrent respectivement le résultat de l'expérience sur les bases *Spambase*, *Ionosphere* et *Parole/musique*.

Nous n'avons pas superposé les critères d'erreur et de séparabilité, dont la comparaison n'a pas de sens. Les critères d'erreurs sont représentés dans les figures supérieures, et comparés à l'erreur *Leave-one-out*, tandis que les critères de séparabilité apparaissent dans les figures inférieures, et ne sont comparables entre eux que par la seule localisation du maximum (leurs valeurs ne sont pas comparables). On remarquera que le logarithme de σ est utilisé ici en abscisses. Le tableau de droite qui accompagne chaque couple de figures indique dans la seconde colonne le temps de calcul t (en secondes), la valeur $\hat{\sigma}$ qui minimise ou maximise le critère, ainsi que la valeur de l'erreur *Leave-one-out* pour le $\hat{\sigma}$ estimé.

Sur les figures, la lettre C suivant les noms des critères d'alignement et de séparabilité (KCS) indique que la procédure d'ajustement de la matrice de Gram est appliquée. « Séparabilité reg » indique quant à lui l'application de la procédure de régularisation pour éviter la divergence du critère de séparabilité lors de la phase d'optimisation. La validation croisée est appliquée sur une division de la base d'apprentissage en dix sous-ensembles.

On constate en premier lieu que la validation croisée constitue sans conteste l'estimateur le plus précis de l'erreur LOO, mais au prix d'une complexité prohibitive. Nous rappelons que la validation croisée ne permet pas d'appliquer de recherche par optimisation et implique donc nécessairement une recherche par maillage. Nous précisons par ailleurs que si, en théorie, le calcul de l'erreur LOO est beaucoup plus coûteux que celui de la validation croisée, nous avons exploité ici l'implémentation optimisée de Joachims [113], tandis que la validation croisée n'est calculée que par apprentissages



Critère	t	$\hat{\sigma}$	**LOO**
LOO	2.59	1.76	7.9%
Validation croisée	1.94	0.91	8.8%
Erreur $\xi\alpha$	0.19	1.13	8.1%
Nb Vec Sup	0.19	1.41	8.0%
Rayon-Marge	0.27	0.73	10.0%
Alignement	0.11	0.47	11.9%
Alignement C	0.1	1.41	8.0%
Séparabilité	0.1	4.26	13.2%
Séparabilité reg	0.1	1.41	8.0%
Séparabilité C	0.1	1.13	8.1%

FIGURE 4.8 – Comparaison des valeurs optimales $\hat{\sigma}$ du paramètre σ estimées par chaque critère sur la base *Spambase*, et de l'erreur *Leave-One-Out* pour ces valeurs, avec $C = 1$.

successifs de SVM, ce qui explique le rapport relativement faible entre les temps de calcul de ces deux erreurs.

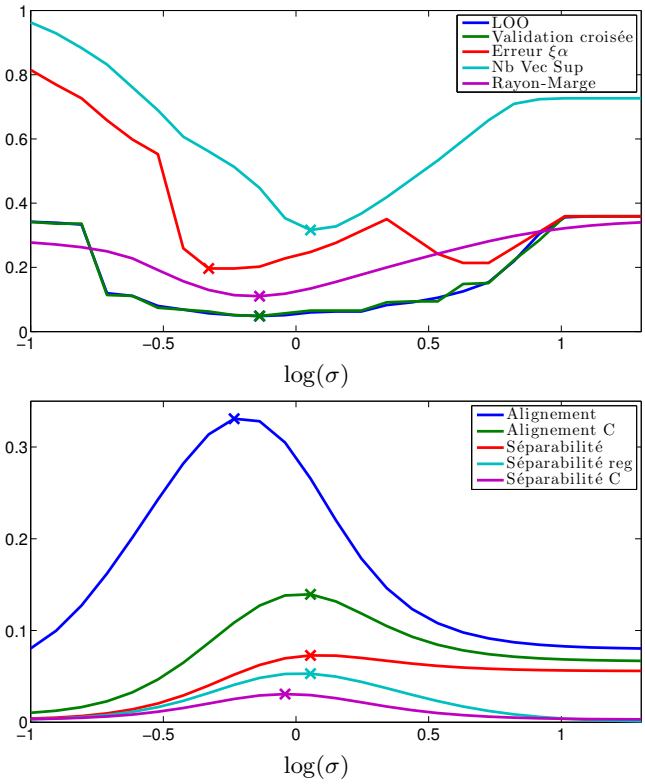
Le nombre de vecteurs de support (Nb Vec Sup) et l'erreur $\xi\alpha$ sont moins gourmants en temps de calcul et constituent bien des bornes supérieures à l'erreur LOO mais on constate en pratique que ces deux bornes sont bien trop larges et surtout que le point minimum est peu corrélé au minimum de l'erreur LOO. De plus, ces critères nécessitent tous deux l'apprentissage d'un SVM et ne sont pas non plus dérivables.

On observe par contre que le critère Rayon-Marge fournit une borne à l'erreur LOO plus resserrée et donc la valeur minimisante est cette fois proche du minimum idéal, sauf dans le cas *Spambase*, où l'estimation est moins précise. Le calcul de la matrice de Gram, ainsi que du rayon R, explique le coût additionnel (le temps de calcul est multiplié par un facteur 1,5) du Rayon-Marge par rapport aux critères précédents.

Les critères proposés (synthétisés dans la partie inférieure des tableaux de résultats) se distinguent nettement des précédents par leur coût en temps de calcul réduit (de l'ordre d'un facteur 3 par rapport au Rayon-Marge), qui résulte de l'absence d'une phase d'apprentissage de SVM dans le calcul. Le calcul de la matrice de Gram est la seule opération coûteuse pour ces critères. L'écart reste cependant moins marqué sur la base *Ionosphere* du fait de sa taille très réduite (les coûts annexes constants y sont donc prédominants).

On remarque que les deux critères proposés ont chacun des limites dans leur forme originelle. Ainsi le critère d'alignement sous-évalue systématiquement la valeur $\hat{\sigma}$ optimale par rapport au minimum de l'erreur LOO, ce qui se traduit par une augmentation raisonnable de l'erreur LOO. En revanche, le critère de séparabilité est ici victime du phénomène de divergence sur le noyau gaussien RBF décrit précédemment, si bien que sur les bases *Spambase* et *parole/musique*, le point maximisant se trouve largement surestimé et produit ainsi une erreur LOO encore plus importante que l'alignement.

La prise en compte du facteur C par la procédure d'ajustement proposée se révèle déterminante sur les deux critères. Sur les bases *Spambase* et *Parole/musique*, on observe ainsi que les valeurs $\hat{\sigma}$ estimées sont très proches des minima de l'erreur LOO (l'égalité stricte, quand elle est observée, est la conséquence du maillage et ne peut être considérée que comme un indice de proximité). Le



Critère	t	$\hat{\sigma}$	**LOO**
LOO	0.2	0.73	4.8%
Validation croisée	0.2	0.73	4.8%
Erreur $\xi\alpha$	0.02	0.47	5.7%
Nb Vec Sup	0.02	1.13	6.0%
Rayon-Marge	0.04	0.73	4.8%
Alignement	0.03	0.58	5.1%
Alignement C	0.03	1.13	6.0%
Séparabilité	0.03	1.13	6.0%
Séparabilité reg	0.02	1.13	6.0%
Séparabilité C	0.03	0.91	5.1%

FIGURE 4.9 – Comparaison des valeurs optimales $\hat{\sigma}$ du paramètre σ estimées par chaque critère sur la base *Ionosphere*, et de l'erreur *Leave-One-Out* pour ces valeurs, avec $C = 1$.

résultat en termes d'erreur est moins évident sur la base *Ionosphere* mais la proximité du minimum demeure cependant encourageante. La procédure de régularisation du critère de séparabilité corrige également le biais déviant mais dans une moindre mesure que l'ajustement.

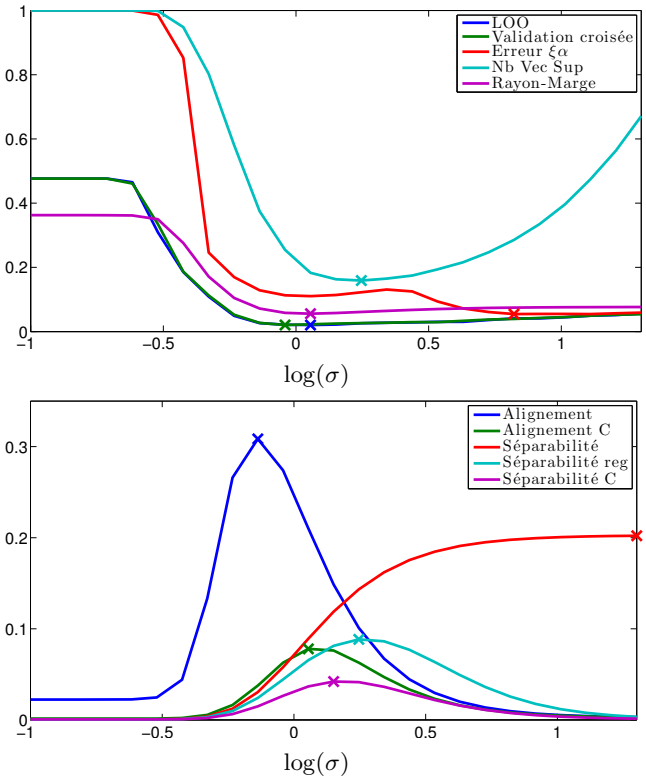
Nous concluons cette expérience par l'application du même protocole sur la base *Lymphoma*. Dans ce cas le facteur C n'est pas fixé d'avance et prend à chaque itération la valeur optimale C_{def} définie par Joachims (voir section 4.5.1). C'est également cette valeur qui est utilisée pour l'ajustement des matrices de Gram. Les résultats sur la base *Lymphoma* sont synthétisés dans la figure 4.11.

La base *Lymphoma* constitue un test plus difficile puisqu'elle contient très peu d'exemples (96) et un nombre élevé de descripteurs (4026) fortement corrélés entre eux. Ceci a pour conséquence de rendre l'allure de la courbe de l'erreur $\xi\alpha$ plus chaotique (dont le minimum à la borne inférieure du maillage, $\hat{\sigma} = 0.1$ est complètement erroné) et réduit considérablement l'ampleur du minimum de l'erreur LOO (vers $\hat{\sigma} = 3.4$) au delà duquel l'erreur ne croît pas vraiment, ce qui complique la tâche pour les autres critères. Ce phénomène est d'ailleurs également dû à la variabilité du facteur C. En effet, l'adaptation du facteur C aux paramètres du noyau (ici le paramètre σ) permet de réduire l'influence de ces derniers sur les performances des SVM. En pratique, donc, si l'usage de C_{def} est préférable, l'affinage des paramètres se trouve alors plus sensible.

Ainsi le critère Rayon-Marge rencontre bien son minimum dans le « bassin » minimal de l'erreur LOO, mais ce minimum reste éloigné du minimum idéal, sans pour autant trop pénaliser les performances. De même, du fait du nombre réduit d'exemples, on constate que l'erreur par validation croisée est plus bruitée que dans les cas précédents et oscille autour de la valeur LOO, tout en restant un bon estimateur.

Les remarques sur les critères proposés demeurent globalement les mêmes. Néanmoins, malgré la divergence apparente des critères de séparabilité (avec ou sans ajustement), il est difficile de juger cette dernière puisque la valeur de σ la plus élevée du maillage produit en pratique une erreur LOO minimale, due au « bassin » de l'erreur LOO que nous avons mentionné. On note cependant que dans ce cas encore l'ajustement de la matrice de Gram permet d'améliorer sensiblement les performances du critère d'alignement.

On constate enfin que les rapports de temps de calcul entre les critères différent par rapport aux

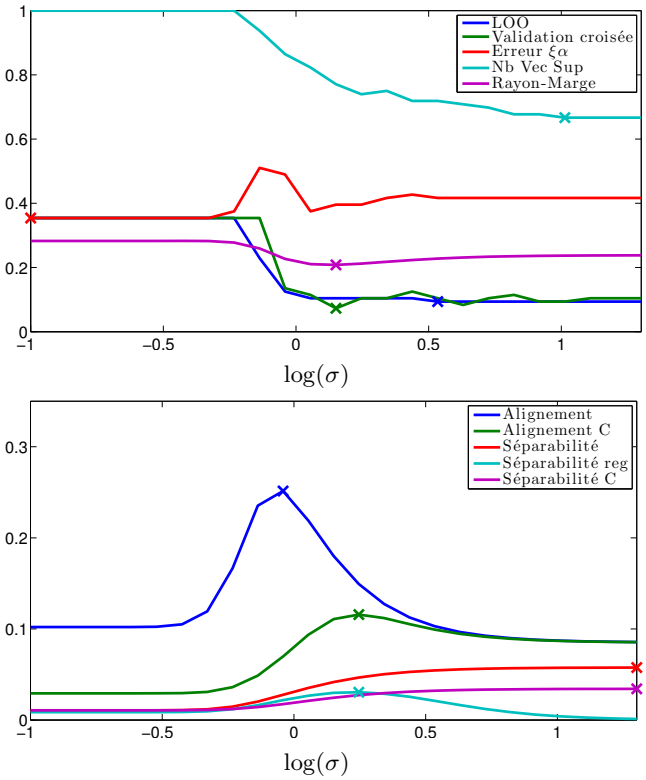


Critère	t	$\hat{\sigma}$	**LOO**
LOO	4.03	1.13	2.0%
Validation croisée	12.5	0.91	2.1%
Erreur $\xi\alpha$	1.53	6.63	4.1%
Nb Vec Sup	1.53	1.76	2.6%
Rayon-Marge	1.99	1.13	2.0%
Alignement	0.64	0.73	2.6%
Alignement C	0.63	1.13	2.0%
Séparabilité	0.62	20.00	5.7%
Séparabilité reg	0.63	1.76	2.6%
Séparabilité C	0.62	1.41	2.2%

FIGURE 4.10 – Comparaison des valeurs optimales $\hat{\sigma}$ du paramètre σ estimées par chaque critère sur la base *Parole/musique*, et de l'erreur *Leave-One-Out* pour ces valeurs, avec $C = 1$.

cas précédent, à cause du nombre élevé de descripteurs qui a une influence directe sur la complexité du calcul de la matrice de Gram. Or, celle-ci n'est calculée qu'une fois dans l'implémentation de Joachims pour le calcul de l'erreur LOO, tandis qu'elle est recalculée à chacune des dix itérations de la validation croisée, ce qui explique que cette dernière soit plus coûteuse dans notre expérience. De même, l'évaluation de la valeur C_{def} est pénalisée par la dimension des exemples, ce qui explique la différence mesurée entre les critères d'alignement et de séparabilité avec et sans ajustement de la matrice de Gram.

Nous avons ainsi introduit plusieurs critères de sélection de noyau, jusqu'ici inusités dans le domaine de la classification audio, de manière à soustraire au processus d'apprentissage la phase habituelle de recherche par maillage. Nous avons montré la pertinence des critères d'Alignement et de Séparabilité des Classe sur des exemples concrets, par rapport aux autres méthodes de la littérature, tout en montrant le gain apporté par notre proposition d'ajustement des matrices de Gram.



Critère	t	$\hat{\sigma}$	**LOO**
LOO	0.59	3.42	9.4%
Validation croisée	0.85	1.41	10.4%
Erreur $\xi\alpha$	0.09	0.10	35.4%
Nb Vec Sup	0.08	10.31	9.4%
Rayon-Marge	0.17	1.41	10.4%
Alignement	0.11	0.91	12.5%
Alignement C	0.19	1.76	10.4%
Séparabilité	0.11	20.00	9.4%
Séparabilité reg	0.11	1.76	10.4%
Séparabilité C	0.20	20.00	9.4%

FIGURE 4.11 – Comparaison des valeurs optimales $\hat{\sigma}$ du paramètre σ estimées par chaque critère sur la base *Lymphoma*, et de l'erreur *Leave-One-Out* pour ces valeurs, avec $C = C_{def}$.