# TP4 et TP5 – Classification par SVM

Le but de ces TP est de tester une autre méthode de classification que celle du TP3, sur les mêmes données. Lancez le script `exercice_0`, qui affiche les caractéristiques de *compacité* et de *contraste* des images de l'ensemble d'apprentissage du TP3, correspondant aux classes « mélanomes » et « fibromes ». Ces données (`X_app` et `Y_app` dans `donnees_carac.mat`) ne sont pas *linéairement séparables*, mais il suffit d'en retirer quelques-unes pour qu'elles le deviennent (`X_app_filtre` et `Y_app_filtre` dans `donnees_carac_filtrees`).

## Contexte : données linéairement séparables - formulation « primale »

Soit $\mathbf{X}_{\text{app}} = (\mathbf{x}_i)_{i\in\{1,\ldots,n\}}$ un ensemble de $n$ points du plan, constitué de deux classes $\omega_1$ et $\omega_2$ linéairement séparables, dont les *étiquettes*, notées $y_i$, valent 1 (pour les fibromes) ou $-1$ (pour les mélanomes). L'équation cartésienne d'une droite $\mathcal{D}$ du plan s'écrit :

$$\mathbf{w}^\top\mathbf{x} - c = 0 \tag{1}$$

où le vecteur non nul $\mathbf{w}$ est orthogonal à $\mathcal{D}$, où $\mathbf{x}$ désigne un point du plan et où $c$ est un paramètre réel. Comme les deux demi-plans limités par $\mathcal{D}$ sont définis par $\mathcal{D}_1 = \{\mathbf{x}\in\mathbb{R}^2, \mathbf{w}^\top\mathbf{x} - c \leq 0\}$ et $\mathcal{D}_2 = \{\mathbf{x}\in\mathbb{R}^2, \mathbf{w}^\top\mathbf{x} - c \geq 0\}$, on peut imposer la contrainte suivante à toute droite $\mathcal{D}$ constituant un *séparateur linéaire* de $\omega_1$ et $\omega_2$ :

$$y_i\left(\mathbf{w}^\top\mathbf{x}_i - c\right) > 0, \qquad \forall\, i \in [\![1,n]\!] \tag{2}$$

Parmi l'infinité de séparateurs linéaires vérifiant la contrainte (2), le SVM est celui qui maximise le carré de la distance minimale des points $\mathbf{x}_i \in \mathbf{X}_{\text{app}}$ à $\mathcal{D}$, ce qui s'écrit :

$$\max_{\mathbf{w}\in\mathbb{R}^2, c\in\mathbb{R}} \left\{ \min_{\mathbf{x}_i\in\mathbf{X}_{\text{app}}} \left\{ \frac{(\mathbf{w}^\top\mathbf{x}_i - c)^2}{\|\mathbf{w}\|^2} \right\} \right\} \equiv \max_{\mathbf{w}\in\mathbb{R}^2, c\in\mathbb{R}} \left\{ \frac{1}{\|\mathbf{w}\|^2} \min_{\mathbf{x}_i\in\mathbf{X}_{\text{app}}} \left\{ (\mathbf{w}^\top\mathbf{x}_i - c)^2 \right\} \right\} \tag{3}$$

D'autre part, l'équation cartésienne (1) de $\mathcal{D}$ est inchangée si $\mathbf{w}$ et $c$ sont multipliés par un même coefficient strictement positif. On peut donc choisir ce coefficient de telle sorte que, pour les points $\mathbf{x}_i$ les plus proches de $\mathcal{D}$, qui sont appelés *vecteurs de support*, on ait exactement $y_i\left(\mathbf{w}^\top\mathbf{x}_i - c\right) = 1$. Dès lors, la contrainte (2) peut être réécrite :

$$y_i\left(\mathbf{w}^\top\mathbf{x}_i - c\right) - 1 \geq 0, \qquad \forall\, i \in [\![1,n]\!] \tag{4}$$

La valeur minimale de $(\mathbf{w}^\top\mathbf{x}_i - c)^2$ vaut alors 1 et le problème (3) se simplifie en :

$$\max_{\mathbf{w}\in\mathbb{R}^2, c\in\mathbb{R}} \left\{ \frac{1}{\|\mathbf{w}\|^2} \right\} \equiv \min_{\mathbf{w}\in\mathbb{R}^2, c\in\mathbb{R}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 \right\} \tag{5}$$

qui constitue un problème de *minimisation quadratique*, sous les contraintes linéaires (4) de type inégalités. Ces problèmes de minimisation quadratique sous contraintes (de types égalités et/ou inégalités) peuvent être résolus de manière efficace par la fonction `quadprog` de Matlab (`help quadprog`). Les inconnues du problème doivent être concaténées en un vecteur $\widetilde{\mathbf{w}} = [\,\mathbf{w}^\top, c\,]^\top \in \mathbb{R}^3$, et le problème reformulé sous forme « canonique » :

$$\left\{ \begin{array}{l} \displaystyle\min_{\widetilde{\mathbf{w}}\in\mathbb{R}^3} \left\{ \frac{1}{2}\widetilde{\mathbf{w}}^\top\mathbf{H}\widetilde{\mathbf{w}} \right\} \\ \quad\text{s.c.} \quad \mathbf{A}\widetilde{\mathbf{w}} \leq \mathbf{b} \end{array} \right. \tag{6}$$

Cette formulation pour résoudre le problème (3) est appelée formulation « primale ».

# Exercice 1 : données linéairement séparables - formulation « duale »

Une autre façon de résoudre le problème (4) + (5) consiste à introduire le *lagrangien* associé, qui dépend non seulement de $\mathbf{w}$ et de $c$, mais également de $n$ *multiplicateurs de Lagrange*, notés $\alpha_i \in \mathbb{R}$, correspondant aux $n$ contraintes linéaires (4) :

$$\begin{aligned}\mathcal{L}(\mathbf{w}, c, \alpha_1, \ldots, \alpha_n) &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left[y_i \left(\mathbf{w}^\top \mathbf{x}_i - c\right) - 1\right] \\ &= \frac{1}{2}\mathbf{w}^\top \mathbf{w} - \mathbf{w}^\top \sum_{i=1}^{n} \alpha_i \, y_i \, \mathbf{x}_i + c \sum_{i=1}^{n} \alpha_i \, y_i + \sum_{i=1}^{n} \alpha_i \end{aligned} \tag{7}$$

Comme les contraintes (4) sont de type $\geq 0$, les multiplicateurs $\alpha_i$ doivent vérifier la contrainte suivante :

$$\alpha_i \geq 0, \qquad \forall\, i \in [\![1, n]\!] \tag{8}$$

De plus, les seuls indices $i$ pour lesquels $\alpha_i > 0$ sont ceux des vecteurs de support, là où $y_i \left(\mathbf{w}^\top \mathbf{x}_i - c\right) - 1 = 0$. Les conditions d'optimalité du premier ordre de $\mathcal{L}$, relativement à $\mathbf{w}$ et à $c$, s'écrivent, respectivement :

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i \, y_i \, \mathbf{x}_i \tag{9}$$

$$\sum_{i=1}^{n} \alpha_i \, y_i = 0 \tag{10}$$

La *fonction duale* du lagrangien $\mathcal{L}$, qui ne dépend que des $\alpha_i$, s'obtient en réinjectant (9) et (10) dans (7) :

$$\overline{\mathcal{L}}(\alpha_1, \ldots, \alpha_n) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \, y_i \, \mathbf{x}_i^\top \mathbf{x}_j \, y_j \, \alpha_j + \sum_{i=1}^{n} \alpha_i \tag{11}$$

Cette fonction étant quadratique mais concave, il faut rechercher son maximum en résolvant un nouveau problème d'optimisation quadratique sous contraintes : contraintes (10) de type égalités + contraintes (8) de type inégalités. En introduisant le vecteur $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]^\top$, la forme canonique de ce problème s'écrit :

$$\left\{ \begin{array}{l} \displaystyle\max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \left\{ -\frac{1}{2}\, \boldsymbol{\alpha}^\top \mathbf{H}\, \boldsymbol{\alpha} + \mathbf{f}^\top \boldsymbol{\alpha} \right\} \\ \qquad \text{s.c.} \left\{ \begin{array}{l} \mathbf{A}_{\text{eq}}\, \boldsymbol{\alpha} = 0 \\ \boldsymbol{\alpha} \geq \mathbf{0}_{\mathbb{R}^n} \end{array} \right. \end{array} \right. \tag{12}$$

Écrivez la fonction `estim_param_SVM_dual`, appelée par le script `exercice_1_dual`, permettant de résoudre le problème (12) par un appel à la fonction `quadprog`.

**Conseils de programmation :**

- Attention au fait que (12) est un problème de *maximisation*, et non de minimisation.
- Une fois trouvés les multiplicateurs de Lagrange, les vecteurs de support `X_VS` sont faciles à identifier, puisque ce sont les points $\mathbf{x}_i$ dont l'indice $i$ est tel que $\alpha_i > 0$ (utilisez ici un seuil à $10^{-6}$ pour ce test).
- Le vecteur $\mathbf{w}$ se déduit de (9), où la somme peut être restreinte aux indices des vecteurs de support.
- Enfin, pour calculer $c$, il suffit par exemple de prendre un vecteur de support $\mathbf{x}_i$ d'étiquette $y_i$, qui vérifie l'égalité $y_i \left(\mathbf{w}^\top \mathbf{x}_i - c\right) - 1 = 0$, soit $c = \mathbf{w}^\top \mathbf{x}_i - y_i$.
- Le dernier paramètre de sortie de la fonction est le code de retour de `quadprog` (« exitflag »), qui vaut 1 lorsque la résolution converge. Vérifiez que cela n'est pas le cas avec les données `X_app` et `Y_app`.

Complétez ensuite la fonction `classification_SVM` qui doit classer les individus $\mathbf{x}_i \in \mathbf{X}_{\text{app}}$ à partir de l'équation (2). Afin de se retrouver avec des valeurs 1 et $-1$ dans le vecteur de prédiction `Y_pred`, il suffit alors de résoudre :

$$y_i = \text{signe}\left(\mathbf{w}^\top \mathbf{x}_i - c\right), \qquad \forall\, i \in [\![1, n]\!] \tag{13}$$

Lancez enfin le script `exercice_1_dual` et vérifiez que vous retrouvez bien le même séparateur linéaire pour les formulations primale et duale sur les deux figures.

## Exercice 2 : données non linéairement séparables - noyau gaussien

Il est rare que des données non filtrées soient linéairement séparables. Pour pallier ce problème, on peut appliquer aux points $\mathbf{x}_i$ une transformation non linéaire, notée $\phi$, de $\mathbb{R}^2$ dans un espace $\mathcal{E}$ de plus grande dimension. Dans cet espace, on cherche un *hyperplan* séparateur, ayant pour équation cartésienne :

$$\mathbf{w}^\top \phi(\mathbf{x}) - c = 0 \tag{14}$$

où $\mathbf{w} \in \mathcal{E}$ et $c \in \mathbb{R}$, devant vérifier les contraintes suivantes :

$$y_i \left(\mathbf{w}^\top \phi(\mathbf{x}_i) - c\right) > 0, \qquad \forall\, i \in [\![1, n]\!] \tag{15}$$

La formulation duale présente alors un avantage important sur la formulation primale car l'extension du problème (6) nécessite de changer d'espace de recherche, puisque l'inconnue $\mathbf{w} \in \mathcal{E}$, alors que l'inconnue $\boldsymbol{\alpha} \in \mathbb{R}^n$ du problème (12) est indépendante de l'espace $\mathcal{E}$. L'extension de la fonction duale (11) s'écrit dorénavant :

$$\overline{\mathcal{L}}(\alpha_1, \dots, \alpha_n) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i\, y_i\, \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)\, y_j\, \alpha_j + \sum_{i=1}^{n} \alpha_i \tag{16}$$

qui fait bien intervenir deux vecteurs de $\mathcal{E}$, mais **seulement par le biais de leur produit scalaire** $\phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$. Le « coup du noyau » (*kernel trick*), qui n'est pas une spécificité des SVM, consiste à remplacer ce produit scalaire par une fonction $K$, appelée *fonction noyau*, ce qui permet de réécrire (16) sous la forme suivante :

$$\overline{\mathcal{L}}(\alpha_1, \dots, \alpha_n) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i\, y_i\, K(\mathbf{x}_i, \mathbf{x}_j)\, y_j\, \alpha_j + \sum_{i=1}^{n} \alpha_i \tag{17}$$

Le noyau qui sera utilisé par la suite est le *noyau gaussien*, où le paramètre $\sigma$ représente un écart-type :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \tag{18}$$

Écrivez la fonction `estim_param_SVM_noyau`, permettant de rechercher le maximum de la fonction $\overline{\mathcal{L}}$, définie par (17) et (18), sous les mêmes contraintes que celles du problème (12). Seule la matrice $\mathbf{H}$ sera modifiée.

**Conseils de programmation :**

- Commencez par calculer la *matrice de Gram*, dont l'élément courant `G(i,j)` est égal à $K(\mathbf{x}_i, \mathbf{x}_j)$.
- Comme la fonction $\phi$ n'est pas explicitement connue, $\mathbf{w}$ ne peut pas être calculé explicitement. D'ailleurs, il ne fait pas partie des paramètres de sortie de la fonction `estim_param_SVM_noyau`.
- En revanche, $c$ peut être calculé en reportant l'expression (9) de $\mathbf{w}$ dans l'égalité $y_i \left(\mathbf{w}^\top \mathbf{x}_i - c\right) - 1 = 0$ et en utilisant à nouveau le noyau $K$ pour remplacer les produits scalaires, et où la somme peut être restreinte aux indices $j$ des vecteurs de support :

$$c = \sum_{j=1}^{n} \alpha_j\, y_j\, K(\mathbf{x}_j, \mathbf{x}_i) - y_i \tag{19}$$

Complétez ensuite la fonction `classification_SVM_avec_noyau` qui doit classer les individus $\mathbf{x}_i \in \mathbf{X}_{\text{app}}$ à partir de l'équation (15). Afin de se retrouver avec des valeurs 1 et $-1$ dans le vecteur de prédiction `Y_pred`, il suffit alors de résoudre :

$$y_i = \text{signe}\left(\sum_{j=1}^{n} \alpha_j\, y_j\, K(\mathbf{x}_j, \mathbf{x}_i) - c\right), \qquad \forall\, i \in [\![1, n]\!] \tag{20}$$

Remplissez enfin la fonction `maximisation_classification_SVM_noyau` qui doit, entre autres, retourner le paramètre $\sigma^*$ qui maximise la classification de ***l'ensemble de test***, suite à une estimation des paramètres du SVM sur ***l'ensemble d'apprentissage***. Lancez successivement le script `exercice_2` pour obtenir les courbes d'optimisation, puis le script `exercice_2bis` pour visualiser la classification optimale.

## Exercice 3 : données non linéairement séparables - marge souple

Une autre manière de classer des données non linéairement séparables par SVM utilise le concept de « marge souple » (*soft marging*), qui revient à assouplir les contraintes $y_i \left(\mathbf{w}^\top \mathbf{x}_i - c\right) - 1 \geq 0$ . Cela peut se faire en introduisant de nouvelles variables, appelées « variables de ressort » et notées $\xi_i$ pour modifier la borne inférieure de la façon suivante :

$$y_i \left(\mathbf{w}^\top \mathbf{x}_i - c\right) - 1 \geq -\xi_i, \qquad \forall\, i \in [\![1, n]\!] \tag{21}$$

avec les contraintes suivantes pour être moins stricte que $\geq 0$ dans l'inégalité (21) :

$$\xi_i \geq 0, \qquad \forall\, i \in [\![1, n]\!] \tag{22}$$

En prenant en compte les deux contraintes précédentes, le nouveau problème prend alors la forme :

$$\min_{\substack{\mathbf{w} \in \mathbb{R}^2, c \in \mathbb{R} \\ (\xi_1, \ldots, \xi_n) \in \mathbb{R}^n}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + \lambda \sum_{i=1}^{n} \xi_i \right\} \tag{23}$$

où $\lambda$ est un paramètre permettant de contrôler le poids associé au terme de régularisation $\sum_{i=1}^n \xi_i$ dans la fonction à minimiser. La formulation duale de ce problème nécessite d'introduire $n$ nouveaux multiplicateurs $\beta_i$ dans le lagrangien $\mathcal{L}$, correspondant aux nouvelles contraintes (22) :

$$\begin{aligned} \mathcal{L}_2(\mathbf{w}, c, \alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n) &= \frac{1}{2} \|\mathbf{w}\|^2 + \lambda \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \left[ y_i \left(\mathbf{w}^\top \mathbf{x}_i - c\right) - 1 + \xi_i \right] - \sum_{i=1}^{n} \beta_i\, \xi_i \\ &= \frac{1}{2} \mathbf{w}^\top \mathbf{w} - \mathbf{w}^\top \sum_{i=1}^{n} \alpha_i\, y_i\, \mathbf{x}_i + c \sum_{i=1}^{n} \alpha_i\, y_i + \sum_{i=1}^{n} \alpha_i + \sum_{i=1}^{n} (\lambda - \alpha_i - \beta_i)\, \xi_i \end{aligned} \tag{24}$$

Les contraintes sur les nouveaux multiplicateurs sont similaires à celles des $\alpha_i$, car (22) est de type « supérieur ou égal » :

$$\beta_i \geq 0, \qquad \forall\, i \in [\![1, n]\!] \tag{25}$$

Les conditions d'optimalité du premier ordre de $\mathcal{L}_2$, relativement aux variables $\xi_i$, s'écrivent simplement :

$$\lambda - \alpha_i - \beta_i = 0, \qquad \forall\, i \in [\![1, n]\!] \tag{26}$$

en complément des autres conditions d'optimalité introduites dans la formulation duale. Avec les inégalités $\alpha_i \geq 0$ issues de la formulation duale, ajoutée à (25) et (26), la contrainte d'inégalité de (24) se simplifie en :

$$0 \leq \alpha_i \leq \lambda, \qquad \forall\, i \in [\![1, n]\!] \tag{27}$$

La forme canonique du problème d'optimisation (24) devient alors :

$$\left\{ \begin{array}{l} \displaystyle\max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \left\{ -\frac{1}{2}\, \boldsymbol{\alpha}^\top \mathbf{H}\, \boldsymbol{\alpha} + \mathbf{f}^\top \boldsymbol{\alpha} \right\} \\ \quad \text{s.c.} \left\{ \begin{array}{l} \mathbf{A}_{\text{eq}}\, \boldsymbol{\alpha} = 0 \\ \boldsymbol{\alpha} \geq \mathbf{0}_{\mathbb{R}^n} \\ \boldsymbol{\alpha} \leq \lambda_{\mathbb{R}^n} \end{array} \right. \end{array} \right. \tag{28}$$

Écrivez la fonction `estim_param_SVM_marge`, permettant de rechercher le maximum de la fonction $\overline{\mathcal{L}}$, définie par (24), sous les contraintes décrites dans (28).

**Conseils de programmation :**

- Les vecteurs de support `X_VS` sont encore les points $\mathbf{x}_i$ d'indice $i$ tel que $\alpha_i > 0$, et le vecteur $\mathbf{w}$ se déduit encore de (9), en restreignant la somme aux indices des vecteurs de support.
- En revanche, pour calculer $c$, **on ne doit pas utiliser n'importe quel vecteur de support !** Pour un vecteur de support d'indice $i$ tel que $\alpha_i < \lambda$, d'après (26), le multiplicateur associé à la variable ressort $\xi_i$ vérifie $\beta_i > 0$. Par conséquent, cette variable doit vérifier $\xi_i = 0$. Comme $\mathbf{x}_i$ est un vecteur de support, on sait également que (21) est une égalité, donc on peut affirmer que $y_i \left(\mathbf{w}^\top \mathbf{x}_i - c\right) - 1 = 0$.

La classification du SVM avec marge souple s'effectue avec la fonction `classification_SVM` que vous avez codée précédemment. Remplissez enfin la fonction `maximisation_classification_SVM_marge` qui doit, entre autres, retourner le paramètre $\lambda^*$ qui maximise la classification de ***l'ensemble de test***, suite à une estimation des paramètres du SVM sur ***l'ensemble d'apprentissage***. Lancez successivement le script `exercice_3` pour obtenir les courbes d'optimisation, puis le script `exercice_3bis` pour visualiser la classification optimale.

## Exercice 4 - combinaison du noyau et de la marge souple (facultatif)

Pour obtenir le meilleur des deux méthodes, il est possible de les combiner. En remplaçant le produit scalaire $\mathbf{x}_i^\top \mathbf{x}_j$ par l'expression avec le noyau $K(\mathbf{x}_i, \mathbf{x}_j)$ et en ajoutant la régularisation sur les variables de ressort $\xi_i$, on se retrouve à nouveau à résoudre un problème (28). À partir des fonctions `estim_param_SVM_noyau` et `estim_param_SVM_marge`, complétez la fonction `estim_param_SVM_noyau_marge` qui doit trouver les paramètres pour le problème combiné.

Par la suite, en utilisant la fonction de classification `classification_SVM_avec_noyau`, Remplissez enfin la fonction `maximisation_classification_SVM_noyau_marge` qui doit, entre autres, retourner le couple de paramètres $(\sigma^*, \lambda^*)$ qui maximise la classification de ***l'ensemble de test***, suite à une estimation des paramètres du SVM sur ***l'ensemble d'apprentissage***. Lancez successivement le script `exercice_4` pour obtenir les courbes d'optimisation, puis le script `exercice_4bis` pour visualiser la classification optimale.