TP4 – Classification par SVM

Le but de ce 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 ne sont pas *linéairement séparables*, mais il suffit d'en retirer quelques-unes pour qu'elles le deviennent. Pour ce faire, affectez aux variables `X` et `Y` de ce script, non plus `X_app` et `Y_app`, mais `X_app_filtre` et `Y_app_filtre`.

Exercice 1 : séparateur linéaire (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 ω_1 et ω_2 linéairement séparables, dont les *étiquettes*, notées y_i, valent -1 ou 1. 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 ω_1 et ω_2 :

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

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 :

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

Or, 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 :

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

D'autre part, comme la valeur minimale de $(\mathbf{w}^\top\mathbf{x}_i - c)^2$ vaut alors 1, 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.

Écrivez la fonction `SVM_1`, appelée par le script `exercice_1`, permettant de résoudre le problème (4) + (5). Les 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 » :

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

Conseils de programmation :

- Attention au sens de l'inégalité dans la contrainte de (6), qui diffère du sens de l'inégalité dans (4).
- La fonction `SVM_1` doit retourner les estimations de $\mathbf{w}$ et de c, ainsi que les vecteurs de support `X_VS`, qui sont les points $\mathbf{x}_i \in \mathbf{X}_{\text{app}}$ pour lesquels (4) est une égalité. Néanmoins, plutôt que l'opérateur `==`, il convient d'utiliser un seuil très faible, par exemple 10^{-6}, sur l'écart entre les deux membres de (4).
- Le dernier paramètre de sortie de `SVM_1` 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`.

Exercice 2 : séparateur linéaire (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) :

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

Comme les contraintes (4) sont de type ≥ 0, les multiplicateurs α_i doivent vérifier la contrainte suivante :

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

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 α_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 :

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

Écrivez la fonction `SVM_2`, appelée par le script `exercice_2`, 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 à nouveau un seuil 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 d'identifier un vecteur de support. En effet, pour un vecteur de support $\mathbf{x}_i$ d'étiquette y_i, nous savons que :

$$y_i \left(\mathbf{w}^\top \mathbf{x}_i \, - c\right) - 1 = 0 \tag{13}$$

En guise de validation de ces deux premiers exercices, vérifiez que `exercice_1` et `exercice_2` affichent le même séparateur linéaire, et que les valeurs obtenues pour $\mathbf{w}$ et c sont les mêmes avec ces deux scripts.

Exercice 3 : données non linéairement séparables

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 ϕ, 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 :

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

À ce stade du raisonnement, il est important de remarquer que (14) est une réécriture de (1), dans laquelle $\mathbf{x}$ est remplacé par $\phi(\mathbf{x})$, et (15) une réécriture de (2), dans laquelle $\mathbf{x}_i$ est remplacé par $\phi(\mathbf{x}_i)$. Or, dans la suite du raisonnement, la formulation duale présente un avantage important sur la formulation primale. En effet, l'extension du problème (6) nécessite de changer d'espace de recherche, puisque l'inconnue $\mathbf{w}$ doit être recherchée, dorénavant, dans $\mathcal{E}$, contrairement à l'inconnue du problème (12), qui est encore $\boldsymbol{\alpha} \in \mathbb{R}^n$. Par ailleurs, l'extension de la fonction duale (11) s'écrit :

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

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 :

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

Le noyau la plus souvent utilisé est le *noyau gaussien* :

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

où le paramètre σ représente un écart-type (vous pourrez tester plusieurs valeurs de ce paramètre).

Écrivez la fonction `SVM_3`, appelée par le script `exercice_3`, 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).

Ajoutez, à la fin du script `exercice_3`, le calcul du pourcentage de bonnes classifications des données de test (non linéairement séparables) et jouez sur la valeur du paramètre σ pour maximiser ce pourcentage.

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 ϕ 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 `SVM_3`.
- En revanche, c peut être calculé en combinant (13) et (9). Si $\mathbf{x}_i$ est un vecteur de support d'étiquette y_i :

$$y_i \left\{ \sum_{j=1}^{n} \alpha_j \, y_j \, \mathbf{x}_j^\top \mathbf{x}_i \; - c \right\} - 1 = 0 \tag{19}$$

  ce qui donne, en utilisant à nouveau le noyau K pour remplacer les produits scalaires :

$$c = \sum_{j=1}^{n} \alpha_j \, y_j \, K(\mathbf{x}_j, \mathbf{x}_i) - y_i \tag{20}$$

  où la somme peut être restreinte aux indices j des vecteurs de support.