Projet

Le Projet d'Analyse de Données complète l'étude de la classification par SVM du TP4, et propose plusieurs extensions. Il doit être effectué en binôme. Les deux séances encadrées devront être complétées par du travail personnel. La séance de validation (pas encore affichée sur ADE) aura lieu entre le 29 mai et le 2 juin (semaine 22). Elle prendra la forme d'une présentation sur machine de 15 minutes par binôme.

Formulation primale du SVM linéaire (exercice 1 du TP4)

Soit $\mathbf{X}_{\text{app}} = (\mathbf{x}_i)_{i\in\{1,\dots,n\}}$ 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 :

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

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 :

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

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.

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 » :

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

Les vecteurs de support `X_VS` sont les points $\mathbf{x}_i \in \mathbf{X}_{\text{app}}$ pour lesquels (4) est une égalité. 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).

Formulation duale du SVM linéaire (exercice 2 du TP4)

Une autre façon de résoudre le problème (4) + (5) consiste à introduire un *lagrangien*, 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 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, qui est un problème de *maximisation*, et non de minimisation, s'écrit :

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

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 $\alpha_i > 0$. Le vecteur $\mathbf{w}$ se déduit de (9), formule dans laquelle 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 :

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

Extension 1 : SVM à noyau gaussien (exercice 3 du TP4)

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 (12) présente un avantage important sur la formulation primale (6). 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 spécifique aux 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 un *noyau gaussien* d'écart-type σ :

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

Écrivez la fonction `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).

Modifiez le script `exercice_3` afin de calculer le pourcentage de bonnes classifications des données de test, qui sont non linéairement séparables.

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.

Extension 2 : optimisation du SVM à noyau gaussien

En jouant sur la valeur du paramètre σ (écart-type du noyau gaussien), vous observerez que la fonction `quadprog` ne parvient pas toujours à converger. Parmi les valeurs de σ « admissibles », trouvez celle qui maximise le pourcentage de bonnes classifications des données d'apprentissage. Calculez ensuite le pourcentage de bonnes classifications des données de test avec cette valeur optimale de σ, afin d'évaluer la capacité de *généralisation* de ce classifieur.

Extension 3 : SVM linéaire à 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 (4). Cela peut se faire en introduisant de nouvelles variables, appelées « variables de ressort » et notées ξ_i, $i \in \{1, \ldots, n\}$, de façon à remplacer (4) par :

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

Si ces variables vérifient les contraintes suivantes :

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

alors, en effet, les contraintes (21) sont plus souples que (4). Une façon d'éviter que ces nouvelles contraintes soient trop souples consiste à remplacer la formulation primale du SVM (problème (5) + contraintes (4)) par :

$$\left\{ \begin{array}{l} \displaystyle\min_{\substack{\mathbf{w} \in \mathbb{R}^2, c \in \mathbb{R} \\ (\xi_1, \ldots, \xi_n) \in \mathbb{R}^n}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + \lambda \sum_{i=1}^{n} \xi_i \right\} \\ \quad \text{s.c.} \left\{ \begin{array}{l} y_i \left(\mathbf{w}^\top \mathbf{x}_i - c\right) - 1 \geq -\xi_i, \qquad \forall\, i \in \{1, \ldots, n\} \\ \xi_i \geq 0, \qquad \forall\, i \in \{1, \ldots, n\} \end{array} \right. \end{array} \right. \tag{23}$$

où λ est un paramètre permettant de contrôler le poids du terme $\sum_{i=1}^{n} \xi_i$ dans la fonction à minimiser, appelé *terme de régularisation*.

La formulation duale de ce problème nécessite d'introduire n nouveaux multiplicateurs β_i, $i \in \{1, \ldots, n\}$, correspondant aux nouvelles contraintes (22), et à remplacer l'expression (7) du lagrangien $\mathcal{L}$ par :

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

Les contraintes sur les nouveaux multiplicateurs sont similaires à (8), car (22) est de type « supérieur ou égal » :

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

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

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

en complément des autres conditions d'optimalité (9) et (10). En réinjectant (9), (10) et (26) dans (24), la fonction duale de $\mathcal{L}_2$ est à nouveau la fonction $\overline{\mathcal{L}}(\alpha_1, \ldots, \alpha_n)$ définie en (11). La classification par SVM avec marge souple consiste donc à rechercher le *maximum* de $\overline{\mathcal{L}}$, sous les mêmes contraintes (8) et (10) que dans le problème (12), mais avec les nouvelles contraintes (25). Or, combinées à (26), ces contraintes s'écrivent :

$$\alpha_i \leq \lambda, \qquad \forall\, i \in \{1, \ldots, n\} \tag{27}$$

Faites une copie du script `exercice_2`, de nom `exercice_2_souple`, que vous modifierez de manière à classer les données d'apprentissage non filtrées, qui ne sont pas linéairement séparables, par résolution de ce nouveau problème d'optimisation quadratique, en choisissant par exemple $\lambda = 100$.

Comme précédemment, cherchez la valeur optimale de λ, relativement aux données d'apprentissage, puis évaluez la capacité de généralisation de ce classifieur.

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 ξ_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 afffirmer que $y_i \left(\mathbf{w}^\top \mathbf{x}_i - c\right) - 1 = 0$.

Extension 4 : SVM à noyau gaussien et marge souple

Écrivez une nouvelle version du script `exercice_3`, de nom `exercice_3_souple`, en suivant la même approche de marge souple que précédemment, par exemple avec les valeurs suivantes des paramètres : $\sigma = 0.075$ et $\lambda = 10000$. Cherchez à nouveau à optimiser le pourcentage de bonnes classifications des données d'apprentissage en jouant sur ces paramètres, puius évaluez la capacité de généralisation de ce classifieur.

Conseils de programmation :

- La seule chose qui change dans l'appel de la fonction `quadprog`, en comparaison du script `exercice_3`, est la contrainte (27).
- Pour calculer le paramètre c, il faut utiliser le produit scalaire d'un vecteur de support $\mathbf{x}_i$ dont l'indice i est tel que $\alpha_i < \lambda$, auquel cas on sait que $y_i \left(\mathbf{w}^\top \mathbf{x}_i - c\right) - 1 = 0$, et remplacer le produit scalaire $\mathbf{w}^\top \mathbf{x}_i$ par une somme sur les vecteurs de support dans laquelle le produit scalaire peut être remplacé par le noyau gaussien.