TD 4

Exercice 1 – Perceptron

Q 1.1 Rappelez la fonction coût au sens des moindres carrés sur un problème d'apprentissage binaire. Proposer quelques exemples pour montrer que les échantillons correctement classés participent à la fonction coût.

Q 1.2 Quel est la fonction de coût utilisée par l'algorithme du perceptron ?

Q 1.3 En imaginant une fonction f de complexité infinie (capable de modéliser n'importe quelle frontière de décision), tracez à la main la frontière de décision optimale au sens des coûts définis précédemment pour le deux problèmes jouets de la figure 1. Ces frontières sont-elles *intéressantes* ? Quels problèmes se posent ?

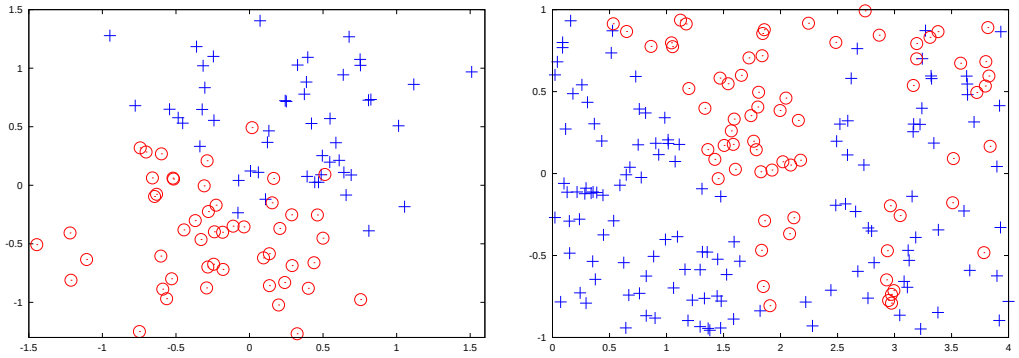

FIGURE 1 – Gaussiennes non séparables linéairement

Q 1.4 Soit $\mathbf{w} = (2, 1)$ le vecteur de poids d'une séparatrice linéaire. Dessinez cette séparatrice dans le plan. Précisez sur le dessin les quantités $< \mathbf{w}, \mathbf{x} >$ par rapport à un exemple $\mathbf{x}$ bien classé et mal classé. Que se passe-t-il pour le produit scalaire dans le cas d'un exemple mal classé avec la mise-à-jour $\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$?

Q 1.5 Comment sont les classifieurs suivants par rapport à celui de la question précédente : $w^1 = (1, 0.5)$, $w^2 = (200, 100)$, $w^3 = (-2, -1)$?

Q 1.6 Montrez que l'algorithme du perceptron correspond à une descente de gradient. La solution est-elle unique ?

Q 1.7 Quel problème peut-il se poser pour certaines valeurs de w ? Comment y remédier ?

Q 1.8 Quelle est la différence entre une descente de gradient stochastique et une descente de gradient batch ? Et mini-batch ?

Q 1.9 Donner un perceptron qui permet de réaliser le `AND` logique entre les entrées binaires x_1 et x_2 (positif si les deux sont à 1, négatif sinon) et un autre pour le `OR` logique.

Exercice 2 – Convergence du Perceptron

On suppose dans cet exercice un ensemble de données $\{\mathbf{x}^i, y^i)\}_{i=0}^N$ linéairement séparable. Nous allons étudier la convergence de l'algorithme du perceptron au fur et à mesure des itérations. Pour cela, nous n'allons considérer que les itérations "utiles", c'est-à-dire celles où une mise-à-jour du vecteur de poids $\mathbf{w}$ est effectuée. On suppose par ailleurs $\mathbf{w}^0 = 0$. On notera par ailleurs $R = max_{i=1}^N \|x^i\|$

Q 2.1 Soit $\gamma > 0$ et $\mathbf{w}^*$ tels que $y_i \frac{<\mathbf{w}^*.\mathbf{x}^i>}{\|\mathbf{w}^*\|} \geq \gamma$ pour tous les exemples du jeu de données. Que signifie géométriquement l'existence de γ et $\mathbf{w}^*$? Si bw^* existe, est-il unique ?

Q 2.2 Donnez un majorant de $\|w^t\|^2$ en fonction de t et de R en utilisant une règle d'induction sur t.

Q 2.3 Donnez un minorant de $< w^t, w^* >$.

Q 2.4 En utilisant l'ingalité de Cauchy-Schwarz $| < u, v > | \leq \|u\|\|v\|$, démontrez le théorième de Novikoff : le nombre d'itérations t de l'algorithme est borné par $\frac{R^2}{\gamma^2}$.

Exercice 3 – Expressivité des séparateurs linéaires

On se place dans l'espace des séparateurs linéaires : $f_{\mathbf{w}}(\mathbf{x}) = \sum_{j=1}^{d} x_j w_j$.

Q 3.1 Quelle est la dimension du vecteur $\mathbf{w}$? Rappeler l'écriture matricielle de $f_{\mathbf{w}}(\mathbf{x})$. Tracer approximativement les frontières optimales en utilisant un modèle linéaire basique sur la figure 1.

Q 3.2 Nous allons augmenter l'expressivité du modèle en étendant l'espace de représentation initial dans le cas 2D : $\mathbf{x} = [x_1, x_2]$. Soit la transformation ϕ suivante : $\phi(\mathbf{x}) = [1, x_1, x_2, x_1^2, x_2^2, x_1 x_2]$, considérons le modèle linéaire $f_{\mathbf{w}}(\mathbf{x}) = \sum_{j=1}^{d} \phi_j(\mathbf{x}) w_j$.

- Quelle est la dimension du vecteur $\mathbf{w}$ dans ce cas ?
- A quoi correspond la projection ϕ ?
- Retracer les frontières de décision optimales sur la figure en utilisant cette nouvelle représentation.
- Pouvons nous retrouver les frontières linéaires de la question précédente dans ce nouvel espace ? Dans l'affirmative, donner les coefficients w_j associés.

Q 3.3 Les frontières sont-elles plus *intéressantes* en utilisant la première ou la seconde représentation des données ? Pouvez vous comparer grossièrement l'amplitude de la fonction coût (au sens des moindres carrés par exemple) dans les cas linéaires et quadratiques ? Qu'en déduire ? Sur quel élément vous basez vous pour mesurer la qualité du modèle créé ?

Q 3.4 Afin d'augmenter l'expressivité de notre classe de séparateur, nous nous tournons vers les représentations gaussiennes. L'espace d'entrée est discrétisé par une grille de N^2 points $\mathbf{p}^{i,j}$, puis nous mesurons la similarité gaussienne du point $\mathbf{x}$ par rapport à chaque point de la grille : $s(\mathbf{x}, \mathbf{p}^{i,j}) = Ke^{-\frac{||\mathbf{x}-\mathbf{p}^{\mathbf{i},\mathbf{j}}||^2}{\sigma}}$. La nouvelle représentation de l'exemple est le vecteur contenant pour chaque dimension la similarité de l'exemple à un point de la grille : $\phi(\mathbf{x}) = (s(\mathbf{x}, \mathbf{p}^{1,1}), s(\mathbf{x}, \mathbf{p}^{1,2}), \ldots)$

- Quelle est la dimension du vecteur $\mathbf{w}$?
- Donnez l'expression littérale de la fonction de décision.
- Quel rôle joue le paramètre σ ?

Q 3.5 Introduction (très) pragmatique aux noyaux

- Que se passe-t-il en dimension 3 si nous souhaitons conserver la résolution spatiale du maillage ?
- Afin de palier ce problème, nous proposons d'utiliser la base d'apprentissage à la place de la grille : les points servant de support à la projection seront ceux de l'ensemble d'apprentissage. Exprimer la forme littérale de la fonction de décision dans ce nouveau cadre. Quelle est la nouvelle dimension du paramètre $\mathbf{w}$?
- Que se passe-t-il lorsque σ tend vers 0 ? vers l'infini ? A-t-on besoin de toutes les dimensions de w ou est-il possible de retrouver la même frontière de decision en limitant le nombre de données d'apprentissage ? A quoi cela correspond-il pour $||\mathbf{w}||$?

Exercice 4 – Boosting

Rappel de l'algorithme : on cherche à construire une combinaison de classifieurs faibles $f_T(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$ de manière itérative, de manière à prendre mieux en compte à une itération donnée les erreurs des itérations précédentes. Pour cela, une distribution de poids sur les exemples est considérée et adaptée à chaque itération afin d'augmenter le poids des exemples mal classés, et de baisser le poids des exemples bien classés. Soit $D_t = (w_t(1), \ldots, w_t(n))$ la distribution des poids des exemples au pas t, D_1 correspondant à la distribution uniforme. L'algorithme consiste en l'itération de la procédure suivante :

1. Choisir h_t qui minimise l'erreur selon D_t
2. Calculer l'erreur ϵ_t associé au classifieur h_t selon D_t
3. Fixer $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$
4. Mettre à jour D_{t+1} : $w_{t+1}(i) = \frac{1}{Z_t} w_t(i) e^{(-\alpha_t y_i h_t(x_i))}$, avec Z_t facteur de normalisation

Q 4.1 Introduction

Q 4.1.1 Rappeler le principe et les différences entre le boosting et le bagging. Soit le jeu de données suivant : $Y^+ = \{(-3,-1),(-3,1),(3,-1),(3,1)\}, Y^- = \{(-1,-1),(-1,1),(1,-1),(1,1)\}$. En considérant comme classifieur faible des stumps (fonction de type $\mathbb{1}_{x_i<\theta_i}$, correspondant à un arbre de décision à 2 feuilles), quels sont les deux premiers classifieurs appris ? Sont-ils suffisant pour la classification parfaite ?

Q 4.1.2 Exprimer l'erreur ϵ_t en fonction d'un coût donné $l(x,y)$ et des $w_t(i)$.

Q 4.1.3 Comment varie α_t en fonction de ϵ_t ? Que se passe-t-il pour $w_{t+1}(i)$ si l'exemple i est bien classifié ? mal classifié ?
On va montrer dans la suite que l'algorithme optimise bien l'erreur d'apprentissage. Le principe de la démonstration consiste à montrer que à chaque pas t, l'erreur est borné par $Z = \prod_{j=1}^{t} Z_j$, et que ce produit converge vers 0.

Q 4.2 Nous allons montrer d'abord que le choix de α_t conduit à minimiser Z_t.

Q 4.2.1 Exprimer Z_t et ϵ_t en fonction de $w_t(i)$, α_t et $y_i h_t(x_i)$.

Q 4.2.2 Exprimer $\frac{\partial Z_t}{\partial a_t}$. En déduire la valeur de α_t qui minimise Z_t.

Q 4.2.3 Donner l'expression de Z_t en fonction de ϵ_t pour α_t optimal.

Q 4.2.4 Soit $\gamma_t = \frac{1}{2} - \epsilon_t$. Sachant que $1 - x \leq e^{-x}$, montrer que Z décroît exponentiellement en fonction de t.

Q 4.3 Nous allons montrer maintenant que Z est une borne supérieure de l'erreur 0-1.

Q 4.3.1 Exprimer $w_{t+1}(i)$ en fonction de $h_j(x), \alpha_j(i), Z_j,\ 1 \leq j \leq t$, puis en fonction de $f_t(x_i)$. En déduire une expression de $\sum_i w_t(i)$ en fonction des Z_j et $y_i f_t(x_i)$, puis une expression de $Z = \prod_j Z_j$ en fonction de $y_i f_t(x_i)$

Q 4.4 Montrez que l'erreur $0-1$ est bornée par le coût exponentiel $l(x,y) = e^{(-yf(x))}$. En déduire que Z est un majorant de l'erreur $0-1$.

Q 4.4.1 Conclure sur la décroissance exponentielle de l'erreur.