TD/TME

Exercice 1 – Retour sur Adaboost

Cet exercice permet de retrouver les résultats proposés dans :
Robert E. Shapire. *Theoritical Views of Boosting.* EuroCOLT'99, 1999

1. Décrire succinctement l'algorithme d'Adaboost vu en cours
2. Quel rôle joue la distribution D_t ?
3. Comment est obtenue l'hypothèse finale H ? A quoi correspondent les différents coefficients dans la formulation de H ?
4. Montrer que :
$$D_{final}(i) = \frac{1}{m\Pi_t Z_t} \exp(-y_i H(x_i))$$
5. Montrer ensuite que l'erreur empirique de H est bornée par :
$$\frac{1}{m} \exp(-y_i H(x_i)) = \Pi_t Z_t$$
Vous pourrez dessiner cette fonction afin de trouver plus facilement la réponse.
6. Dans le cas où $\varepsilon_t = P_{i\tilde{D}_t}(h_t(x_i) \neq y_i) = \sum_{i:h_t(x_i)\neq y_i} D_t(i)$, montrer que :
$$Z_t = 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$$
7. Citer les qualités principales de cet algorithme en vue du projet à réaliser.
8. Coder sur feuille une première version de l'algorithme AdaBoost.

Exercice 2 – Perceptron linéaire à seuil

Q 2.1 Un classifieur à deux classes, $C1$,$C2$, opère sur des objets de dimension $d = 2$: $X = \begin{bmatrix} x_{11} & x_{12} \\ \mathbf{x}_i & \\ x_{N1} & x_{N2} \end{bmatrix}$, avec $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^2$ et utilise la fonction discriminante g :

$$\mathbf{x}_i \mapsto g(\mathbf{x}_i) = w_1 x_{i1} + w_2 x_{i2} - \theta,\ \theta \text{ est donné}$$

$\mathbf{x}_i$ est mis dans la classe C_1 si $g(\mathbf{x}_i) > 0$ et dans la classe C_2 si $g(\mathbf{x}_i) < 0$.

1. Quelle est l'équation de la frontière de décision ?
2. On peut mettre en bijection les points $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \in \mathbb{R}^2$ et $\mathbf{x}' = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}^T \in \mathbb{R}^3$. On construit un classifieur g' de paramètres $w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$ pour traiter les points de $\mathbb{R}^3$: $g'(\mathbf{x}') = \mathbf{x}'w$. Quelle valeur faut-il donner à w pour que les deux classifeurs soient équivalents ?
NB : dans l'ensemble du TD, on considère $\mathbf{x}_i$ comme un vecteur ligne et w comme un vecteur colonne.

3. On veut coder $+1$ les objets attribués à la classe C_1 et -1 les objets attribués à la classe C_2 ; quelle fonction F faut-il utiliser pour que la composée $F \circ g$ réalise ce codage ?

   N.B. dans la terminologie des réseaux de neurones, $\mathbf{x}$ est une entrée, g le potentiel de $\mathbf{x}$, θ le seuil, F la fonction d'activation et $\{-1, +1\}$ les sorties ; le classifieur précédent est un perceptron linéaire à seuil, sans couche cachée, à une cellule de sortie.

Q 2.2

On veut utiliser le perceptron précédent pour implémenter le ET logique à deux arguments ; pour cela : on identifie $Vrai$: $+1$ et $Faux$: -1 ; une entrée est un couple $x1, x2 \in \{-1, +1\}$ et une sortie un élément de $\{-1, +1\}$.

1. Que vaut ici $\mathcal{X}$? Montrer qu'un perceptron implémente le ET logique si et seulement si $w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$ vérifie un certain système d'inéquations.
2. Trouver une solution du système précédent. Est-elle unique ? Démontrer votre résultat à l'aide d'un schéma en 2D puis en 3D (dans $\mathbb{R}^2$ puis dans $\mathbb{R}^3$).
3. Mêmes questions pour le OU logique.
4. Montrer que le OU Exclusif ne peut pas être implémenté par un perceptron linéaire à seuil.

Exercice 3 – Apprentissage du perceptron

N.B. Les notations sont les mêmes que dans l'exercice précédent.

On dispose d'une base de N exemples (observations), $\{\mathbf{x}_i\}_{i=1,\ldots,N}$, dont les classes sont connues ; la classe de $\mathbf{x}_i$ est notée $d(\mathbf{x}_i)$. On utilise l'algorithme suivant (une des variantes de l'algorithme du perceptron) pour apprendre automatiquement la valeur des paramètres, c-à-d du vecteur w : (ε est un nombre positif ; $\mathbf{x}^T$ est le transposé de $\mathbf{x}$)

Algorithme 1 : Algorithme d'apprentissage du perceptron

Entrée : $\{\mathbf{x}_i, d(\mathbf{x}_i)\}_{i=1,\ldots,N}$;
Initialisation de w : $w(1)$;
$t = 1$;
repeat
  Tirer aléatoirement un exemple : $\mathbf{x}_i$;
  if $d(\mathbf{x}_i) \times \mathbf{x}_i w(t) \geq 0$ **then**
    $w(t+1) \leftarrow w(t)$
  else
    $w(t+1) \leftarrow w(t) + \varepsilon d(\mathbf{x}_i)\mathbf{x}_i^T$
  $t = t + 1$;
until *(critère d'arrêt satisfait)* ;

Le critère d'arrêt peut être, par exemple qu'il n'y a pas eu d'erreur de classification pendant un certain nombre d'itérations successives.

Q 3.1 Que signifie la condition $d(\mathbf{x}_i) \times \mathbf{x}_i w(t) \geq 0$? Expliquez le principe de l'algorithme.

Q 3.2 Faites tourner l'algorithme sur le problème du OU logique en itérant sur la base d'apprentissage constituée des 4 exemples distincts possibles, avec successivement pour valeur initiale w(0) :

1. $w(0) = (0; 0; 0)$;
2. $w(0) = (1; 1; 1)$;
3. $w(0) = (1; -1; 1)$.

(Prendre $\varepsilon = 1$) Représenter graphiquement l'évolution de la frontière de décision d'itération en itération.

Q 3.3 Dérouler l'algorithme sur le problème du ET logique avec $w(0) = (2; 2; 2)$.

Q 3.4 On suppose qu'il existe $w^\star$ classant parfaitement tous les exemples (séparabilité linéaire). On considère une itération de l'algorithme où l'exemple courant, $x[= x(t)]$, vérifie $x \in C_1$ mais est mal classé par le vecteur $w[= w(t)]$ courant, qui est alors modifié pour devenir $w^\dagger[= w(t + 1)]$.

1. Vérifier qu'alors : $w^\star \cdot \mathbf{x} > 0; w \cdot \mathbf{x} < 0; w^\dagger = w + \varepsilon\mathbf{x}$
2. Montrer que : $\|w^\dagger - w^\star\|^2 \leq \|w - w^\star\|^2 + \varepsilon[\varepsilon\|\mathbf{x}\|^2 - 2w^\star \cdot \mathbf{x}]$
3. On pose $m = \min_{i=1,\ldots,N} w^\star \cdot \mathbf{x}_i$ et $M = \max_{i=1,\ldots,N} \|\mathbf{x}_i\|^2$. Montrer qu'en prenant $\varepsilon = \frac{m}{M}$, on obtient l'inégalité $\|w^\dagger - w^\star\|^2 \leq \|w - w^\star\|^2 - \frac{m^2}{M}$.
4. Admettant que l'inégalité précédente est aussi valable dans le cas symétrique ($\mathbf{x} \in C_2$ et mal classé), montrer qu'il y a un nombre fini d'itérations où w est modifié ; en déduire que l'algorithme peut être arrêté au bout d'un nombre fini d'itérations.

Exercice 4 – Alternative algorithmique pour apprendre le perceptron

Nous présentons ici l'algorithme ADALINE (suivant la règle dite de Widrow-Hoff)

Algorithme 2 : Algorithme ADALINE

Entrée : $\{\mathbf{x}_i, d(\mathbf{x}_i)\}_{i=1,\ldots,N}$;
Initialisation de w : $w(1)$;
$t = 1$;
repeat
    for $i = 1, \ldots, N$ **do**
        $w(t + 1) \leftarrow w(t) - \varepsilon\,(d(\mathbf{x}_i) - \mathbf{x}_i w(t))\,\mathbf{x}_i^T$
    $t = t + 1$;
until *(critère d'arrêt satisfait)* ;

Q 4.1 Repérer quelles sont les dimensions des différentes matrices/vecteurs de l'algorithme. Quelles sont les différences entre les deux algorithmes vus dans ce TD ?

Q 4.2 Montrer que cet algorithme correspond à une descente de gradient. Quelle est la fonction coût minimisée ?

Q 4.3 Imaginer au moins trois critères distincts pour sortir des boucles while de ces deux algorithmes.