

TRAVAUX DIRIG  S N   5 : Perceptron

Stephan CL  MEN  ON <stephan.clemencon@telecom-paris.fr>

Quentin BOUNIOT <quentin.bouniot@telecom-paris.fr>

EXERCICE 1. On se place dans le cadre de la classification binaire, comme au TD 4. On observe cette fois-ci un n -  chantillon $\mathcal{D}_n = \{(X_i, Y_i)\}_{1 \leq i \leq n}$, repr  sentant n copies ind  pendantes du (X, Y) . Pour des param  tres $\beta \in \mathbb{R}^d$ et $\beta_0 \in \mathbb{R}$ on note $f_{\beta, \beta_0} : x \in \mathbb{R}^d \mapsto \beta^\top x + \beta_0$ puis H_{β, β_0} l'hyperplan (affine) d  fini par $H_{\beta, \beta_0} = \{x \in \mathbb{R}^d : f_{\beta, \beta_0}(x) = 0\}$. On consid  re les classifieurs de la forme $\text{sgn} \circ f_{\beta, \beta_0}$ et on suppose que les donn  es sont lin  airement s  parables.

- 1) Soit $x \in \mathbb{R}^d$. Calculer $d(x, H_{\beta, \beta_0})$ o   d est la distance euclidienne.
- 2) On propose comme mesure d'erreur associ  e    une fonction f_{β, β_0} (*Hinge loss*) :

$$\ell_H(f_{\beta, \beta_0}) = - \sum_{\{i: Y_i f_{\beta, \beta_0}(X_i) < 0\}} Y_i f_{\beta, \beta_0}(X_i).$$

Comparer avec l'  criture de la perte $\ell_{0/1}$ classique utilis  e habituellement.

- 3) V  rifier que l'on peut toujours se ramener    la situation o   $\beta_0 = 0$. On supposera dor  navant que c'est le cas.
- 4) L'algorithme du perceptron consiste    optimiser en β la perte ℓ_H avec une approche de type descente de (sous-)gradient stochastique. On pourra introduire un param  tre de taux d'apprentissage η (*learning rate*), et l'on notera β^k , le coefficient obtenu lorsque l'on a commis la k^e erreur.
Discuter d'initialisations possibles pour β^0 , et proposer les   tapes successives pour mettre    jour β au vue des donn  es.

On va prouver le r  sultat suivant : "Si les deux classes sont strictement s  parables par un hyperplan sur l'  chantillon d'apprentissage alors le perceptron s'arr  te en un nombre fini d'  tapes". On supposera dans la suite que les X_i sont presque-s  urement de norme (euclidienne) born  e par $B > 0$.

- 5) Pour tout $\beta \in \mathbb{R}^d$ on d  finit $\gamma(\beta) := \min_{1 \leq i \leq n} Y_i \beta^\top X_i$. Interpr  ter ce param  tre et en d  duire une formulation explicite de l'hypoth  se de s  parabilit   lin  aire. On pose $\gamma^* := \max_{\|\beta\|=1} \gamma(\beta)$. V  rifier son existence. Quel est son signe ?
- 6) Par simplicit   on initialise l'algorithme avec $\beta^0 = 0$. Montrer alors que le perceptron s'arr  te en au plus $\left(\frac{B}{\gamma^*}\right)^2$ mises    jour.

Indication. Borner $\|\beta^k\|$ pour $k \geq 1$ avant la derni  re mise    jour.

Correction Exercice 1

Dans toute la correction, $\beta \neq 0$ et on note $P = \{i : Y_i = +1\}$, $N = \{i : Y_i = -1\}$.

1) Le plus court chemin vers $H := H_{\beta, \beta_0}$ est dirig   selon la normale β : on cherche un point de H de la forme $x - \lambda\beta$. La condition $\beta^\top (x - \lambda\beta) + \beta_0 = 0$ admet l'unique solution $\lambda^* = f_{\beta, \beta_0}(x) / \|\beta\|^2$, et on pose $p(x) := x - \lambda^*\beta \in H$ (projection orthogonale sur H).

L'infimum est donc atteint en le projeté orthogonal $p(x)$, et

$$d(x, H_{\beta, \beta_0}) = |\lambda^*| \|\beta\| = \frac{|\beta^\top x + \beta_0|}{\|\beta\|}.$$

En particulier, si $\|\beta\| = 1$, $|f_{\beta, \beta_0}(x)|$ est exactement la distance de x à l'hyperplan.

2) En posant $z_i := Y_i f_{\beta, \beta_0}(X_i)$ (marge fonctionnelle) et $\phi(z) := \max(0, -z) = -z \mathbb{1}_{\{z < 0\}}$, les deux pertes s'écrivent

$$\ell_{0/1}(f_{\beta, \beta_0}) = \sum_{i=1}^n \mathbb{1}_{\{Y_i \neq \text{sgn}(f_{\beta, \beta_0}(X_i))\}} = \sum_{i=1}^n \mathbb{1}_{\{z_i < 0\}} \quad \ell_H(f_{\beta, \beta_0}) = \sum_{i=1}^n \phi(z_i).$$

(Normaliser par $1/n$ ne change ni les minimiseurs ni l'algorithme : la constante se reporte dans le taux d'apprentissage.)

- La perte $\ell_{0/1}$ compte chaque erreur pour 1. Elle ne dépend de β qu'à travers les signes des z_i : elle est constante par morceaux, de gradient nul presque partout, non convexe, et sa minimisation est en général NP-difficile. On la retrouverait en remplaçant f par $\text{sgn} \circ f$ dans ℓ_H .
- La perte ℓ_H pénalise chaque erreur par $|z_i| = \|\beta\| d(X_i, H_{\beta, \beta_0})$ (question 1), i.e. proportionnellement à sa distance à la frontière. Elle est continue et convexe en (β, β_0) : ϕ est un maximum de deux fonctions affines, donc convexe, et $(\beta, \beta_0) \mapsto z_i$ est linéaire.

Comme ϕ n'est pas dérivable en 0, on utilisera son sous-différentiel : $\partial\phi(z) = \{-1\}$ si $z < 0$, $[-1, 0]$ si $z = 0$, $\{0\}$ si $z > 0$.

3) On ajoute une coordonnée constante aux descripteurs :

$$\tilde{X}_i = \begin{pmatrix} 1 \\ X_i \end{pmatrix} \in \mathbb{R}^{d+1}, \quad \tilde{\beta} = \begin{pmatrix} \beta_0 \\ \beta \end{pmatrix} \in \mathbb{R}^{d+1}, \quad \tilde{\beta}^\top \tilde{X}_i = \beta^\top X_i + \beta_0 = f_{\beta, \beta_0}(X_i).$$

Le classifieur et la perte sont inchangés : chercher un hyperplan affine de $\mathbb{R}^d$ revient à chercher un hyperplan vectoriel de $\mathbb{R}^{d+1}$. La borne sur les normes devient $\|\tilde{X}_i\| \leq \sqrt{1 + B^2}$, ce qui ne change pas l'argument de la question 6. On note désormais $H_{\tilde{\beta}} := H_{\beta, \beta_0}$.

Preuve géométrique (enveloppes convexes). Soient C_P, C_N les enveloppes convexes des deux classes. Un hyperplan sépare les classes si et seulement s'il sépare C_P et C_N . On prend $x^+ \in C_P$, $x^- \in C_N$ à distance minimale (existent car compacts). La boule ouverte de diamètre $[x^-, x^+]$ ne rencontre ni C_P ni C_N (sinon on trouverait des points plus proches). Pour $z \in [x^-, x^+]$, l'hyperplan perpendiculaire à $[x^-, x^+]$ passant par z sépare les données : un point positif du côté de x^- donnerait un segment $[X_i, x^+] \subset C_P$ traversant la boule. En translatant les données par z , on obtient une séparation par un hyperplan vectoriel. C'est un argument d'existence, mais trouver le bon z serait trop coûteux.

4) On minimise $\ell_H(\beta) = \sum_i \ell_i(\beta)$ avec $\ell_i(\beta) = \phi(Y_i \beta^\top X_i)$. La descente de sous-gradient stochastique utilise un seul terme à la fois : $\beta \leftarrow \beta - \eta g_i$ avec $g_i \in \partial \ell_i(\beta)$. Par composition avec l'application linéaire $\beta \mapsto Y_i \beta^\top X_i$, on a $g_i = s Y_i X_i$ avec $s \in \partial \phi(Y_i \beta^\top X_i)$:

- point bien classé ($Y_i \beta^\top X_i > 0$) : $s = 0$, β est inchangé ;
- point mal classé ou sur la frontière ($Y_i \beta^\top X_i \leq 0$) : on choisit $s = -1$, d'où $\beta \leftarrow \beta + \eta Y_i X_i$.

Le choix $s = -1$ en $z = 0$ (test ≤ 0 et non < 0) est essentiel. Avec $s = 0$, $\beta = 0$ serait un point fixe, alors qu'il ne classe rien. Avec $s = -1$, l'algorithme ne s'arrête que si $\gamma(\beta) > 0$, i.e. sur une séparation stricte, ce qu'utilise la question 6.

Algorithme 1 : Perceptron

Input : données $(X_i, Y_i)_{1 \leq i \leq n}$, taux $\eta > 0$, initialisation β^0

$\beta \leftarrow \beta^0, k \leftarrow 0$;

repeat

foreach $i \in \{1, \dots, n\}$ (dans un ordre quelconque ou aléatoire) **do**

if $Y_i \beta^\top X_i \leq 0$ **then**

$\beta \leftarrow \beta + \eta Y_i X_i$;

$k \leftarrow k + 1$;

// $\beta^k = \beta$: k -ième erreur

until un passage complet sur les données ne déclenche aucune mise à jour;

Output : β

Initialisation. Le plus simple est $\beta^0 = 0$. On peut aussi initialiser aléatoirement, mais avec un pas constant, une initialisation non nulle peut coûter des mises à jour supplémentaires (voir la question 6). De plus il faut choisir la distribution selon laquelle l'initialisation sera faite et ses paramètres. On peut également faire varier η , par exemple par le nombre d'erreurs.

5) $\gamma(\beta) = \min_i Y_i \beta^\top X_i = (\min_{i \in P} \beta^\top X_i) \wedge (-\max_{i \in N} \beta^\top X_i)$ est la marge fonctionnelle du pire point. Comme $\gamma(c\beta) = c\gamma(\beta)$ pour $c > 0$ sans que l'hyperplan change, on se restreint à $\|\beta\| = 1$. Dans ce cas, $|\beta^\top X_i| = d(X_i, H_\beta)$ (question 1), et :

- si $\gamma(\beta) > 0$, tous les points sont strictement bien classés et $\gamma(\beta)$ est la distance du point le plus proche à l'hyperplan (*marge géométrique*);
- si $\gamma(\beta) = 0$, aucun point n'est mal classé mais au moins un est sur l'hyperplan;
- si $\gamma(\beta) < 0$, au moins un point est mal classé, et $-\gamma(\beta)$ est la distance à H_β du plus mal classé.

L'hypothèse de séparabilité linéaire s'écrit donc : *il existe* $\beta \in \mathbb{R}^d, \|\beta\| = 1$, *tel que* $\gamma(\beta) > 0$.

Existence de γ^* . La sphère unité $\{\|\beta\| = 1\}$ est compacte (fermée bornée en dimension finie) et γ est continue (minimum fini de fonctions linéaires) : γ y atteint son maximum en un β^* unitaire.

Signe. Si les données sont séparables, $\gamma^* \geq \gamma(\beta) > 0$. Sinon, $\gamma(\beta) \leq 0$ pour tout β , donc $\gamma^* \leq 0$. Ainsi, les données sont linéairement séparables si et seulement si $\gamma^* > 0$, et γ^* est la plus grande marge atteignable par un hyperplan séparateur. Par Cauchy-Schwarz, $\gamma^* \leq B$.

6) Soit β^* unitaire tel que $\gamma(\beta^*) = \gamma^* > 0$, de sorte que $Y_i \beta^{*\top} X_i \geq \gamma^*$ pour tout i . On note i_k le point utilisé à la k -ième mise à jour : $Y_{i_k} \beta^{k-1\top} X_{i_k} \leq 0$ et $\beta^k = \beta^{k-1} + \eta Y_{i_k} X_{i_k}$. L'idée est d'encadrer $\|\beta^k\|$.

Minorant. $\beta^{k\top} \beta^* = \beta^{k-1\top} \beta^* + \eta Y_{i_k} \beta^{*\top} X_{i_k} \geq \beta^{k-1\top} \beta^* + \eta \gamma^*$. En itérant depuis $\beta^0 = 0$, puis par Cauchy-Schwarz ($\|\beta^*\| = 1$) :

$$\|\beta^k\| \geq \beta^{k\top} \beta^* \geq k\eta \gamma^*.$$

Majorant. Comme on ne met à jour que sur une erreur :

$$\|\beta^k\|^2 = \|\beta^{k-1}\|^2 + 2\eta \underbrace{Y_{i_k} \beta^{k-1\top} X_{i_k}}_{\leq 0} + \eta^2 \underbrace{\|X_{i_k}\|^2}_{\leq B^2} \leq \|\beta^{k-1}\|^2 + \eta^2 B^2, \quad \text{d'où} \quad \|\beta^k\|^2 \leq k\eta^2 B^2.$$

Conclusion. $k^2 \eta^2 (\gamma^*)^2 \leq \|\beta^k\|^2 \leq k\eta^2 B^2$, donc tout nombre k de mises à jour effectuées vérifie $k \leq (B/\gamma^*)^2$. Le perceptron s'arrête en un nombre fini d'étapes, sur un β qui sépare strictement les données. ■

Le taux η s'est simplifié : avec $\beta^0 = 0$, tous les β^k sont proportionnels à η . La difficulté est mesurée par le rapport rayon/marge B/γ^* , indépendamment de n et de d (et $(B/\gamma^*)^2 \geq 1$ puisque $\gamma^* \leq B$). C'est une garantie dans le pire des cas. Si les données ne sont pas séparables, l'algorithme ne s'arrête jamais.

Cas général : β^0 quelconque et taux η_k variable. On suppose $\gamma_0 := \gamma(\beta^0) \leq 0$ (sinon aucune mise à jour n'a lieu) et on raisonne sur $\beta^k - \beta^0$. Le même minorant donne $\|\beta^k - \beta^0\| \geq \gamma^* \sum_{\ell \leq k} \eta_\ell$. Pour le majorant,

$$\|\beta^k - \beta^0\|^2 = \|\beta^{k-1} - \beta^0\|^2 + \eta_k^2 \|X_{i_k}\|^2 + 2\eta_k \underbrace{Y_{i_k} \beta^{k-1^\top} X_{i_k}}_{\leq 0} - 2\eta_k \underbrace{Y_{i_k} \beta^0^\top X_{i_k}}_{\geq \gamma_0},$$

d'où $\|\beta^k - \beta^0\|^2 \leq B^2 \sum_{\ell \leq k} \eta_\ell^2 + 2|\gamma_0| \sum_{\ell \leq k} \eta_\ell$. En combinant les deux bornes :

$$1 \leq \left(\frac{B}{\gamma^*}\right)^2 \frac{\sum_{\ell \leq k} \eta_\ell^2}{\left(\sum_{\ell \leq k} \eta_\ell\right)^2} + \frac{2|\gamma_0|}{(\gamma^*)^2} \frac{1}{\sum_{\ell \leq k} \eta_\ell}.$$

- $\eta_k = \eta, \beta^0 = 0 : k \leq (B/\gamma^*)^2$, on retrouve le résultat précédent.
- $\eta_k = \eta, \beta^0 \neq 0 : k \leq (B/\gamma^*)^2 + 2|\gamma_0|/(\eta(\gamma^*)^2)$. Une mauvaise initialisation coûte des mises à jour, d'autant plus que η est petit.
- $\sum_\ell \eta_\ell \rightarrow \infty$ et $\sum_\ell \eta_\ell^2 / (\sum_\ell \eta_\ell)^2 \rightarrow 0$: le membre de droite tend vers 0, donc l'inégalité est violée au-delà d'un certain rang k_0 , et l'algorithme fait au plus k_0 mises à jour.

Enfin, pour plus de détails sur le perceptron, une bonne reference est le livre de Ripley : [\[Ripley, 1996\]](#).

Références

[Ripley, 1996] Ripley, B. (1996). *Pattern recognition via neural networks*. Cambridge University Press. 4