Introduction à la classification

Classification Multi-classe
Théorie de l'apprentissage statistique

Guillaume Wisniewski
guillaume.wisniewski@limsi.fr

Université Paris Sud — LIMSI

Février 2016

Détails pratiques

1. Leçon inaugurale de Yann Lecun au collège de France, jeudi 18h : l'apprentissage profond une révolution en intelligence artificielle
2. sujet de projet en ligne, détails en TP
3. stages
   - Guillaume Wisniewski& Hélène Maynard : traduction de la parole (cf. Skype)
   - Hervé Bredin & Claude Barras (LIMSI) : reconnaissance du locteur, fouille de données dans les série télé
   - voir avec Aurélien pour les possibilités au LRI

Première partie I

Rappel : classification binaire

Cadre

En entrée

- espace d'observations $\mathcal{X}$ (généralement $\mathbb{R}^d$)
- espace de sortie $\mathcal{Y} = \{-1, 1\}$ (ou $\{0, 1\}$)
- ensemble d'apprentissage $\left(\mathbf{x}^{(i)}, y^{(i)}\right)_{i=1}^{N}$

En sortie

- hypothèse de classification : $h : \mathcal{X} \mapsto \mathcal{Y}$

Perceptron — Paramétrisation du problème

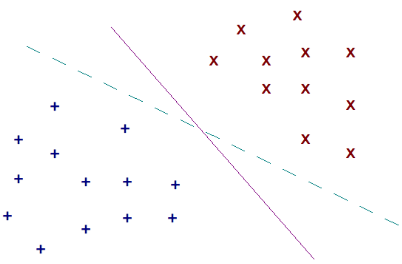
- classe de fonction considérée :

$$h(\mathbf{x}) = \text{sign} \langle \mathbf{x} | \mathbf{w} \rangle \quad (1)$$

$$= \sum_{i=1}^{d} x[i] \cdot w[i] \quad (2)$$

  où $\mathbf{w} \in \mathbb{R}^d$ est un vecteur de paramètres
- recherche d'un hyperplan séparateur entre les deux classes

En image



Algorithme d'apprentissage

- objectif : estimer le vecteur de paramètres à partir d'un ensemble d'exemples

Require: a training set $\left(\mathbf{x}^{(i)}, y^{(i)}\right)_{i=1}^{N}$

```
w ← 0
while there are classification errors do
    for i = 1 to n do
        y = sign(⟨w|x^(i)⟩)
        if y ≠ y^(i) then
            w ← w + ·y^(i) · x^(i)
        end if
    end for
end while
```


Interprétation de la règle de mise à jour

Changement de point de vue

- deux vecteurs de paramètres :
  - $\mathbf{w}_1$ paramètres de la classe positive
  - $\mathbf{w}_{-1}$ paramètres de la classe négative

  avec $\mathbf{w}_1 = -1 \times \mathbf{w}_{-1}$
- règle de décision :

$$y^* = \arg\max_{y \in \{1,-1\}} \langle \mathbf{w}_y | \mathbf{x} \rangle \quad (3)$$

$$= \begin{cases} 1 & \text{si } \langle \mathbf{w}_1 | \mathbf{x} \rangle \geq \langle \mathbf{w}_{-1} | \mathbf{x} \rangle \\ -1 & \text{sinon} \end{cases} \quad (4)$$

- interprétation : le score $\langle \mathbf{w}_y | \mathbf{x} \rangle$ mesure la confiance que l'on a dans l'hypothèse « la classe de $\mathbf{x}$ est y »

Deuxième partie II

Perceptron multi-classe

Cadre

En entrée

- $\mathcal{Y} = \{1, 2, ..., k\}$ $(2 < k < \infty)$
- le reste est inchangé

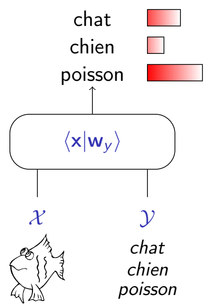
Paramétrisation

- un vecteur de paramètre par classe $\mathbf{w}_y, \forall y \in \mathcal{Y}$
- règle de décision

$$y^* = \arg\max_{y \in \mathcal{Y}} \langle \mathbf{w}_y | \mathbf{x} \rangle \quad (5)$$

$\Rightarrow$ on recherche la classe de plus grand score

Interprétation



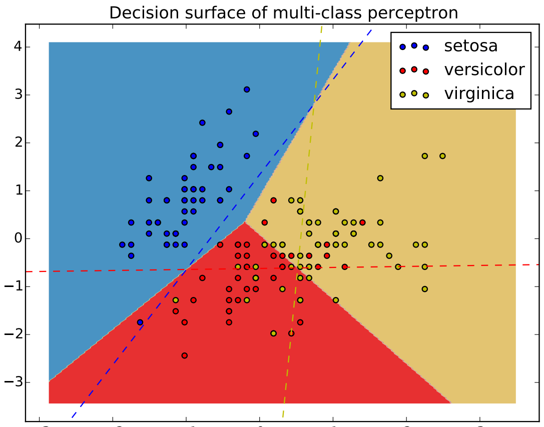
Méthode d'apprentissage

- seul la règle de mise à jour change (lorsqu'une erreur est détectée) :
  - y^* = étiquette prédite pour l'observation $\mathbf{x}^{(i)}$
  - $y^{(i)}$ = étiquette de référence (que l'on aurait dû prédire)
- mise à jour :

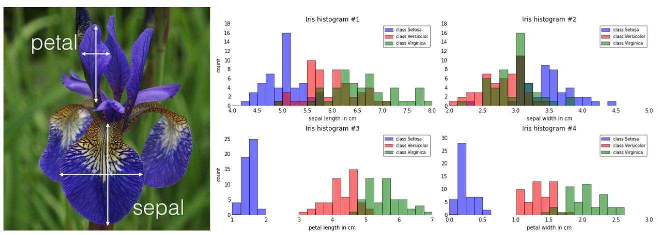
$$\begin{cases} \mathbf{w}_{y^*} & \leftarrow \mathbf{w}_{y^*} - \mathbf{x} \\ \mathbf{w}_{y^{(i)}} & \leftarrow \mathbf{w}_{y^{(i)}} + \mathbf{x} \end{cases} \tag{6}$$

- on renforce l'étiquette que l'on aurait dû prédire et on pénalise celle que l'on a prédite (à tord)

Frontière de décision



La tâche



- Jeu de données Iris
- identifier 3 types d'iris à partir de 4 caractéristiques

Troisième partie III

Évaluation

Principe de l'évaluation

- élément le plus important
- besoin :
  - évaluation quantitative
  - évaluation rapide
  - évaluation répétitive
- fonction de coût :

$$\ell^{0/1}(y, y') = \begin{cases} 1 & \text{si } y \neq y' \\ 0 & \text{sinon} \end{cases} \tag{7}$$

- fonction de score : on « inverse » la fonction de coût

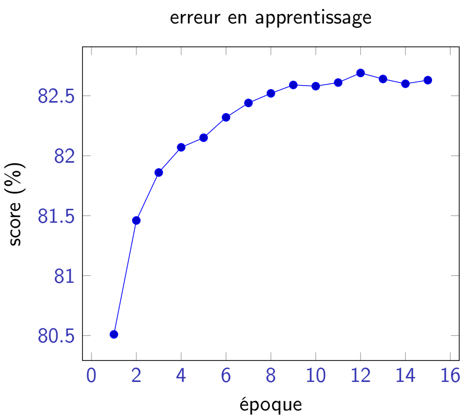
Utilisation

- erreur en apprentissage d'une hypothèse h :

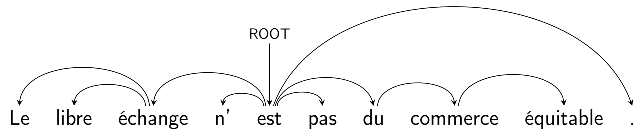
$$\ell^{\text{train}} = \frac{1}{N} \sum_{i=1}^{N} \ell^{0/1}\left(y^{(i)}, h(\mathbf{x}^{(i)})\right) \tag{8}$$

- principal intérêt : évaluer la capacité du système à apprendre
- courbe d'apprentissage : évolution de l'erreur d'apprentissage en fonction du nombre d'itérations / exemples vus

Exemple



Analyse en dépendance



Attention :

- règle de décision : perceptron incrémental et non perceptron
- apprentissage : perceptron moyenné et non perceptron

Limites

Erreur en apprentissage

- trivial de trouver un algorithme qui obtient une erreur d'apprentissage nulle : mode SQL
- erreur optimiste : accès aux particularités des données
- ce qui nous intéresse : prédire l'étiquette d'observation inconnue = généraliser
- mais : toujours à calculer pour s'assurer que l'apprentissage « marche »

Erreur en généralisation

- quel sera le taux d'erreur sur un ensemble de données jamais vues .
- plus généralement : quelle garantie en généralisation peut-on observer à partir de l'observation d'un échantillon fini de données ?

Quatrième partie IV

Introduction à la théorie de l'apprentissage statistique

Apprentissage supervisé

- espace d'observations $\mathcal{X}$ (généralement $\mathbb{R}^d$)
- espace de sortie $\mathcal{Y} = \{-1, 1\}$ (ou $\{0, 1\}$)
- ensemble d'observations étiquetées $\left(\mathbf{x}^{(i)}, y^{(i)}\right)_{i=1}^{N}$
- hypothèse apprise : $h : \mathcal{X} \mapsto \mathcal{Y}$
- fonction de coût $\ell(y, y')$

Aspect essentiel : on a accès à des données étiquetées par un expert

Modèle de génération des données

Définition

- les observations sont générées aléatoirement et de manière indépendante selon une distribution $\mathcal{D}$ fixe et inconnu
- étant donnée une observation un oracle f attribue l'étiquette correspondante

Interprétation

- hypothèses fausses et invérifiables
- nécessaires pour savoir dans quel cas l'apprentissage est possible (essentiellement pour les preuves)
- en pratique : les observations sont « semblables » et on n'est pas dans un cadre « adverse »

Risque fonctionnel

Définition

Manière naturelle d'évaluer la qualité d'une hypothèse h :

$$R[h] = \int_{\mathcal{X}\times\mathcal{Y}} \ell\left(h(\mathbf{x}, y)\right) \cdot \mathrm{d}\mathcal{D}(\mathbf{x}, y)$$
$$= \mathbb{P}_{\mathbf{x}\sim\mathcal{D}}[h(\mathbf{x}) \neq f(\mathbf{x})]$$

dans le cas du coût 0/1

Interprétation

- erreur sur l'ensemble des observations possibles
- probabilité qu'une observation (observée selon une distribution $\mathcal{D}$) soit mal classée
- calculable que si l'on accès à l'ensemble des observations possibles !

Principe ERM (1)

- le risque empirique n'est pas calculable...
- ...mais on peut l'estimer sur l'ensemble d'apprentissage $S = \left(x^{(i)}, y^{(i)}\right)_{i=1}^{m}$:

$$L_S(h) = \frac{\left|\left\{h(\mathbf{x}^{(i)}) \neq y^{(i)}, \forall i \in [\![1, m]\!]\right\}\right|}{m} \tag{9}$$

- moyenne du coût sur l'ensemble d'apprentissage
- synonymes : erreur en apprentissage, risque empirique
- critère de sélection simple : trouver h qui minimise le risque empirique

$\Rightarrow$ Empirical Risk Minimization

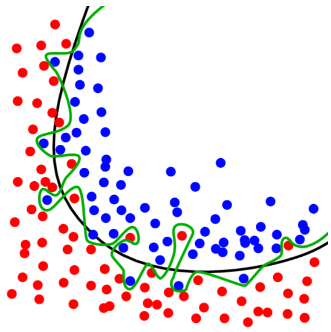
Insuffisance du principe ERM (1)

- supposons que $\mathcal{X}$ soit uniformément distribué à l'intérieur d'un carré de surface 2 ; tous les exemples à gauche du carré sont considérés comme positifs, ceux à droit comme négatifs
- il existe une hypothèse dont l'erreur d'apprentissage est nulle :

$$h_S(x) = \begin{cases} y_i & \text{si } \exists i, x^{(i)} = x \\ 0 & \text{sinon} \end{cases} \tag{10}$$

- c'est l'hypothèse qui sera choisie selon ERM
- son erreur en généralisation est au moins $\frac{1}{2}$

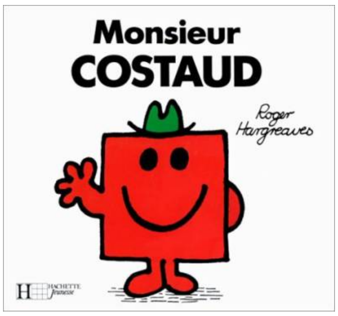
Insuffisance du principe ERM (2)



Conclusion

- faible erreur en apprentissage, mais très mauvaise généralisation
- sur-apprentissage *(overfitting)*
- trop « adapté » au bruit des données d'apprentissage, faible capacité à généraliser

Solution : biais inductif



- limiter le type de fonction recherché
- minimiser le risque empirique dans une classe de fonctions donnée $\mathcal{H}$
- typiquement : fonctions linéaires, programmes Python qui s'écrivent en n octets, ...

$$h^* = \arg\min_{h\in\mathcal{H}} L_S(h) \tag{11}$$

Remarque j'avais déjà utilisé cet argument dans le cas de la régression, mais pour des raisons « calculatoires »

Le compromis fondamental de l'apprentissage

- réduire le nombre de fonctions que l'on considère $\Rightarrow$ limiter la capacité ($\simeq$ expressivité) d'une hypothèse d'apprentissage
- 2 risques :
  - $\mathcal{H}$ est trop riche : sur-apprentissage
  - $\mathcal{H}$ est trop pauvre : sous-apprentissage
- adapter la capacité de la classe de fonctions aux données $\Rightarrow$ objectif fondamental de la théorie de l'apprentissage

Hypothèses simplificatrices

- realizability : il existe une fonction de $h^* \in \mathcal{H}$ dont le risque est nul
  $\Rightarrow$ h^* a une erreur d'apprentissage nulle sur tout échantillon de $\mathcal{D}$
- les données de S sont i.i.d.
- $\mathcal{H}$ est un ensemble fini (p.ex. tous les programmes de taille n)

arguments techniques qui permettent de simplifier les preuves mais dont on peut se passer

Ce qu'il est raisonnable d'espérer

Choix de h à partir d'un échantillon fini S

S est choisi aléatoirement

- peut être non représentatif
- dans ce cas : aucun algorithme ne peut trouver une bonne hypothèse
- il faut que l'algorithme d'apprentissage puisse « échouer » avec une certaine probabilité

S est incomplet :

- on ne verra jamais tous les exemples
- il est illusoire d'espérer un classifieur parfait

$\Rightarrow$ Probably Approximately Correct learning

PAC-learning

- δ : probabilité que le résultat soit faux
- ϵ : taux d'erreur max.
- objectif : l'algorithme d'apprentissage doit garantir avec une probabilité d'au moins $1 - \delta$ que l'hypothèse trouvée a une erreur inférieure à ϵ :

$$\mathbb{P}\left(L_{(\mathcal{D},f)}(h) > \epsilon\right) \leq \delta$$

où $L_{(\mathcal{D},f)}(h) = R[h]$

Résultat

Théorème

Let $\mathcal{H}$ be a finite hypothesis class. Let $\delta \in (0,1)$ and $\epsilon > 0$ and let m be an integer that satisfies

$$m \geq \frac{\log\left(|\mathcal{H}|/\delta\right)}{\epsilon} \tag{12}$$

Then, for any labeling function, f, and for any distribution, $\mathcal{D}$, for which the realizability assumption holds (that is, for some $h \in \mathcal{H}, L_{(\mathcal{D},f)}(h) = 0$), with probability of at least $1 - \delta$ over the choice of an i.i.d. sample S of size m, we have that for every ERM hypothesis, h_S, it holds that :

$$L_{(\mathcal{D},f)}(h_S) \leq \epsilon \tag{13}$$

Interprétation

Ce que l'on a montré

- minimiser l'erreur en apprentissage (sur un ensemble de données finies) permet de minimiser l'erreur en généralisation (sur un ensemble de données infini)...
- dès que l'ensemble d'apprentissage est suffisamment grand

Facteur

D'autant plus d'exemples que

- la classe de fonctions est riche
- la précision demandée est grande