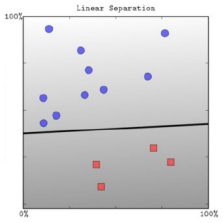
SPLEX
Statistiques pour la classification et fouille de données en génomique
Classification Linéaire Binaire (CLB)

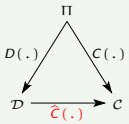
Pierre-Henri WUILLEMIN

DEcision, Système Intelligent et Recherche opérationnelle
LIP6
pierre-henri.wuillemin@lip6.fr
http://webia.lip6.fr/~phw/splex

Classification linéaire binaire (CLB)



Définition (CLB)



- $\mathcal{C} = \{\ominus, \oplus\}$
- $\exists w \in \mathbb{R}^d, w_0 \in \mathbb{R}, \exists f : \mathbb{R} \to \mathcal{C},$

$$\forall x \in \mathbb{R}^d, \widehat{C}(x) = f\left(\sum_{i=1}^{d} w_i \cdot x_i + w_0\right)$$

Le problème d'apprentissage : trouver w, w_0 et f.

Modèles génératifs, modèles discriminants

- Modèles génératifs : classification grâce à une estimation de $P(x, y)$ à partir de Π_a et des connaissances *a priori*.
  - Classifieur bayésien (ML, MAP)
  - Classifieur bayésien naïf
  - Discriminant linéaire de Fisher
- Modèles discriminants : estimation directe des w, w_0 à partir de Π_a.
  - Régression logistique
  - Perceptron
  - SVM

Le classifieur bayésien naïf binaire est un CLB ?

Classifieur bayésien naïf

$$y = \arg\max_{y_i} \left(P(y_i) \cdot \prod_{k=1}^{d} P(x^k \mid y_i) \right)$$

- Ici, $y_0 = \ominus$ et $y_1 = \oplus$.
- Soit $R(x) = \frac{P(\oplus) \cdot \prod_{k=1}^{d} P(x^k \mid \oplus)}{P(\ominus) \cdot \prod_{k=1}^{d} P(x^k \mid \ominus)}$
- Si $R(x) > 1$ alors $\widehat{C}(x) = \widehat{\oplus}$ sinon $\widehat{C}(x) = \widehat{\ominus}$
- Donc $\widehat{C}(x) = \sigma(\log R(x))$ où $\sigma(u) = \begin{cases} -1 & \text{si } u < 0 \\ 0 & \text{si } u = 0 \\ +1 & \text{sinon} \end{cases}$
- Il vient alors

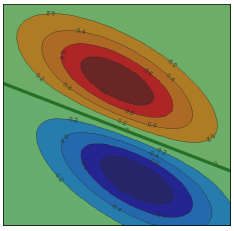
$$\widehat{C}(x) = \sigma\left(\log \frac{P(\oplus)}{P(\ominus)} + \sum_{k=1}^{d} \log \frac{P(x^k \mid \oplus)}{P(x^k \mid \ominus)} \right)$$

Suite évidente dans le cas binomial ($\mathcal{D} = \{\ominus, \oplus\}^d$)).

Discrimination linéaire - cas gaussien

Cadre gaussien

- Modèle : $\widehat{C}(x) = \sigma(g(x)) = \sigma\left(g_{\oplus}(x) - g_{\ominus}(x)\right)$
- Régions de décision : $\forall c \in \left\{\widehat{\ominus}, \widehat{\oplus}\right\}, R_c = \left\{x \in \mathcal{D}, \widehat{C}(x) = c\right\}$
- Frontière de décision : $F = \left\{x \in \mathcal{D}, \widehat{C}(x) = 0\right\}$
- Multinormalité : $\forall c \in \{\ominus, \oplus\}$, $P(x \mid c) \sim \mathcal{N}(\mu_c, \Sigma_c)$



CLB

Si homoscédasticité : $\forall c, \Sigma_c = \Sigma$ alors, la fonction discriminante devient linéaire :

$$g(x) = \left(\mu_{\oplus} - \mu_{\ominus}\right)^t \Sigma^{-1} (x - x_0)$$

avec $x_0 = \frac{1}{2}\left(\mu_{\oplus} + \mu_{\ominus}\right) + \left(\frac{1}{\left(\mu_{\oplus} - \mu_{\ominus}\right)^t \Sigma^{-1} \left(\mu_{\oplus} - \mu_{\ominus}\right)} \log \frac{P(\oplus)}{P(\ominus)}\right)\left(\mu_{\oplus} - \mu_{\ominus}\right)$

Rappels de géométrie

Soit $y(x) = \sum_{i=1}^{d} w_i \cdot x_i + w_0 \Rightarrow \widehat{C}(x) = f(y(x))$, on peut également écrire :

$$y(x) = w' \cdot x + w_0 \qquad \text{avec } y(x) = 0 \text{ l'équation d'un hyperplan } H$$

$\forall a, b \in H, y(a) = y(b) = 0 \Rightarrow y(a) - y(b) = w' \cdot (a - b) = 0$

w est un vecteur normal à H.

Soit $x \notin H$ et x_H sa projection perpendiculaire sur H, $x - x_H$ est donc colinéaire à w,
Soit $r \in \mathbb{R}, x - x_H = r * \frac{w}{||w||}$ où r est la distance de x à H.

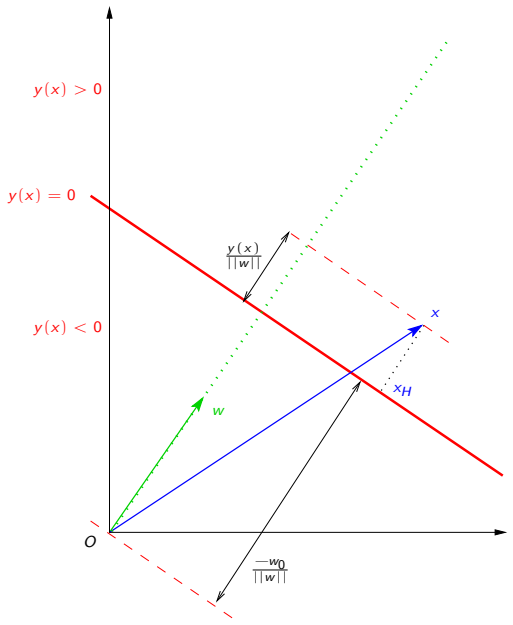
$$x = x_H + r \cdot \frac{w}{||w||}$$

$$w' \cdot x = w' \cdot x_H + r \cdot \frac{w' \cdot w}{||w||} = w' \cdot x_H + r \cdot \frac{||w||^2}{||w||} = w' \cdot x_H + r \cdot ||w||$$

$$y(x) = w' \cdot x + w_0 = w' \cdot x_H + w_0 + r \cdot ||w|| = y(x_H) + r \cdot ||w|| = r \cdot ||w||$$

distance de x à H : $r = \dfrac{y(x)}{||W||}$

Rappels de géométrie



exemple : Hyper-plan séparateurs

La frontière entre les deux classes est donnée par $\sum_{i=1}^{d} w_i \cdot x_i + w_0 = 0$ qui est l'équation d'un hyper-plan.
Comment choisir cet hyper-plan ?

Exemple : CLB par régression linéaire

- Ajuster un modèle linéaire $\hat{I}_k$ pour chaque fonction indicatrice d'une classe k :

$$\forall k \in \{\oplus, \ominus\}, \hat{I}_k(x) = \begin{cases} 1 & \text{si } x \text{ est de classe } k \\ 0 & \text{sinon.} \end{cases}$$

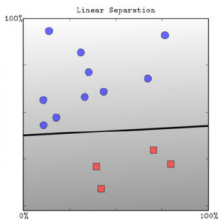
$$\hat{I}_{\oplus}(x) = \beta_{\oplus 0} + \beta'_{\oplus} \cdot x \text{ et } \hat{I}_{\ominus}(x) = \beta_{\ominus 0} + \beta'_{\ominus} \cdot x$$

- Soit un x à classifier : $\widehat{C}(x) = \arg\max_k \hat{I}_k(x) = \sigma(\hat{I}_{\oplus} - \hat{I}_{\ominus})$

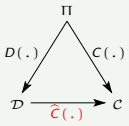
Frontière de décision :

$$\hat{f}_{\oplus}(x) = \hat{f}_{\ominus}(x) \Rightarrow \text{hyperplan} : \begin{cases} w = & \beta_{\oplus} - \beta_{\ominus} \\ \text{et} & \\ w_0 = & \beta_{\oplus 0} - \beta_{\ominus 0} \end{cases}$$

Séparabilité



Définition (CLB)



- $\mathcal{C} = \{\ominus, \oplus\}$
- $\exists w \in \mathbb{R}^d, w_0 \in \mathbb{R}, \exists f : \mathbb{R} \to \mathcal{C}$,

$$\forall x \in \mathbb{R}^d, \widehat{C}(x) = f\left(\sum_{i=1}^{d} w_i \cdot x_i + w_0\right)$$

Le problème d'apprentissage : trouver w, w_0 ($\mathbf{W}$) et f (souvent σ).

Séparabilité sur Π_a

Soit une base de données $\Pi_a = (\mathbf{x}_i, y_i)_{i<N}$ où y_i est la classe de $\mathbf{x}_i$ ($\in -1, +1$).
Π_a est linéairement séparable si il existe un hyperplan d'équation $y(x) = w' \cdot x + w_0 = 0$) tel que

$$\forall i \in \{1, \cdots, N\}, y(\mathbf{x}_i) \cdot y_i > 0 \text{ i.e. } (\mathbf{X} \cdot \mathbf{W}) \times \mathbf{Y} > 0$$

Optimisation de $\mathbf{W}$: moindres carrés


Carl Friedrich Gauss

- $\mathbf{X} \cdot \mathbf{W} - \mathbf{Y}$ est le vecteur des erreurs effectuées en classant Π_a à l'aide de $\mathbf{W}$.
- L'erreur quadratique obtenue sur Π_a se calcule donc comme :

$$e^2(\mathbf{W}) = (\mathbf{X} \cdot \mathbf{W} - \mathbf{Y})' \cdot (\mathbf{X} \cdot \mathbf{W} - \mathbf{Y})$$

- Minimiser cette erreur en annulant le gradient donne :

$$\mathbf{W}^* = (\mathbf{X}' \cdot \mathbf{X})^{-1} \cdot \mathbf{X}' \cdot \mathbf{Y} = \mathbf{X}^{\dagger} \cdot \mathbf{Y}$$

$\mathbf{X}^{\dagger} = (\mathbf{X}' \cdot \mathbf{X})^{-1} \cdot \mathbf{X}'$ est la pseudo-inverse de X.

Cette méthode souffre de plusieurs problèmes :

- Instabilité numérique (pour des $\mathbf{X}$ de grande taille principalement),
- Manque de robustesse pour des distributions larges de classes.

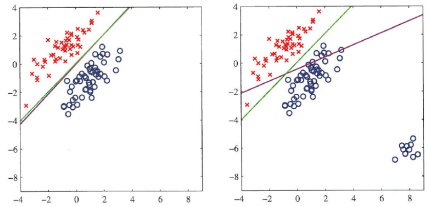


Figure 4.4 The left plot shows data from two classes, denoted by red crosses and blue circles, together with the decision boundary found by least squares (magenta curve) and also by the logistic regression model (green curve), which is discussed later in Section 4.3.2 The right-hand plot shows the corresponding results obtained when extra data points are added at the bottom left of the diagram, showing that least squares is highly sensitive to outliers, unlike logistic regression.

From : *Pattern Recognition and Machine Learning* – C.Bishop – p186

Discrimants de Fisher : séparation entre les classes

- On note que $y = \mathbf{w}' \cdot \mathbf{x}$ correspond à la projection de $\mathbf{x}$ (de dimension $d+1$) sur la droite vectorielle $\mathbf{w}$.
- Soit $\mathbf{M}_{\oplus} = \frac{1}{N_{\oplus}} \sum_{i \in \oplus} \mathbf{X}_i$ et $\mathbf{M}_{\ominus} = \frac{1}{N_{\ominus}} \sum_{i \in \ominus} \mathbf{X}_i$
- On peut alors utiliser $\Delta_{\mathbf{w}} = \mathbf{w}' \cdot (\mathbf{M}_{\oplus} - \mathbf{M}_{\ominus})$ comme mesure de la séparation des classes selon $\mathbf{w}$.

Afin de supprimer l'influence sur $\Delta_{\mathbf{w}}$ de la norme de $\mathbf{w}$, on peut soit normaliser $\mathbf{w}$, soit utiliser $\frac{\Delta_{\mathbf{w}}}{||\mathbf{w}||}$ comme mesure.

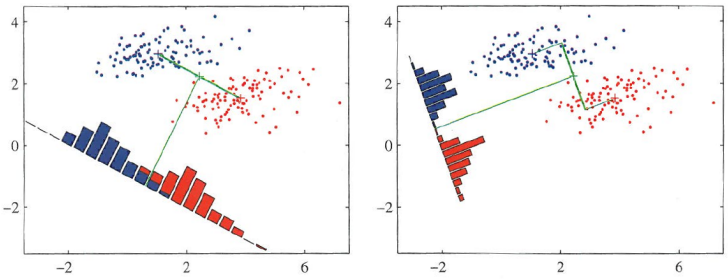


Figure 4.6 The left plot shows samples from two classes (depicted in red and blue) along with the histograms resulting from projection onto the line joining the class means. Note that there is considerable class overlap in the projected space. The right plot shows the corresponding projection based on the Fisher linear discriminant, showing the greatly improved class separation.

From : *Pattern Recognition and Machine Learning* – C.Bishop – p188

- La séparation des classes n'est intéressante qu'en fonction de la dispersion de chaque classe, i.e. $\forall k \in \oplus, \ominus, s_k = \sum_{i \in k} (y_i - \mathbf{w}' \cdot \mathbf{M}_k)^2$ les variances *intra-classe*.

De la régression linéaire vers la régression logistique

Régression linéaire

$$\hat{y}(x) = w' \cdot x + w_0$$

Frontière de séparation : hyperplan d'équation $y(x) = w' \cdot x + w_0 = 0$

En réutilisant MAP pour décider :

$$\hat{y} = \arg \max_{c \in \{\oplus, \ominus\}} p(c \mid x)$$

On ne peut pas ajuster linéairement une probabilité : une droite n'est pas bornée par $[0, 1]$.
Idée : La frontière de décision correspond à

$$p(\oplus \mid x) = p(\ominus \mid x) \iff \frac{p(\oplus \mid x)}{p(\ominus \mid x)} = 1 \iff \log \frac{p(\oplus \mid x)}{p(\ominus \mid x)} = 0$$

On peut renforcer l'idée que la frontière est un hyperplan (CLB) par :

Régression logistique

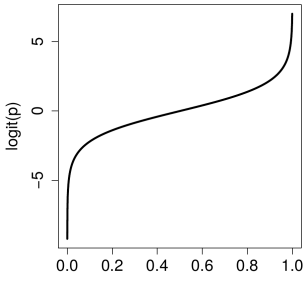
$$\exists w, w_0, \log \frac{p(\oplus \mid x)}{p(\ominus \mid x)} = w' \cdot x + w_0$$

fonction logit

On peut écrire $\log \frac{p(\oplus|x)}{p(\ominus|x)} = \log \frac{p(\oplus|x)}{1-p(\oplus|x)}$

Fonction logit (*log-odds*)

$$\text{logit}(p) = \log \frac{p}{1-p}$$



La fonction logit est non bornée et donc peut être ajuster linéairement.

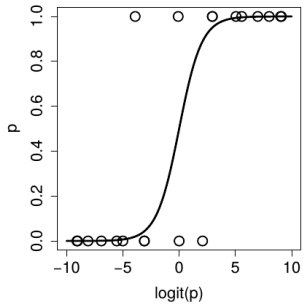
$$\text{logit}(p) = w' \cdot x + w_0 \iff \frac{p}{1-p} = e^{w' \cdot x + w_0} \iff p = \frac{e^{w' \cdot x + w_0}}{1 + e^{w' \cdot x + w_0}}$$

Modèle de la régression logistique

$$p(\oplus \mid x) = \frac{e^{w' \cdot x + w_0}}{1 + e^{w' \cdot x + w_0}} \text{ et } p(\ominus \mid x) = \frac{1}{1 + e^{w' \cdot x + w_0}}$$

Utilisation de la régression logistique

Soit une base $\Pi_a = (X_i, Y_i)$ avec $Y_i \in \{\oplus, \ominus\}$,
On peut calculer pour chacun $w' \cdot x + w_0$,
et donc calculer $p(\oplus \mid x) = \frac{e^{w' \cdot x + w_0}}{1 + e^{w' \cdot x + w_0}}$.



Estimation des paramètres w, w_0

Comment calculer les valeurs de w et w_0 de la régression logistique ?

- Moindre carrés ? Impossible car les erreurs ne sont pas distribuées suivant une loi normale : Elle est quasi nulle quand p proche de 0 ou 1 et plus importante quand $p \approx 0.5$.

- Utilisation du **Maximum de Vraisemblance** :
  - Exprimer la vraisemblance $L(X; w, w_0)$ pour w et w_0,
  - Essayer de maximiser la vraisemblance
  - En annulant la dérivée mais pas de forme exacte de la dérivée.
  - Utiliser une méthode approchée : *Algorithme de Newton-Raphson*.

- Soit une base de données $(X, Y)_{i \leq N}$. Avec $y_i = 1$ si $\oplus$ et 0 si $\ominus$.
  - $\forall i, L(x_i; w, w_0) = y_i \cdot p(x_i|\oplus) + (1 - y_i) \cdot p(x_i|\ominus)$
  - Or si $\log \frac{p(\oplus|x)}{p(\ominus|x)} = w' \cdot x + w_0$ alors $\exists \beta, \beta_0, \log \frac{p(x|\oplus)}{p(x|\ominus)} = \beta' \cdot x + \beta_0$
  - $p(x \mid \oplus) = \frac{e^{\beta' \cdot x + \beta_0}}{1 + e^{\beta' \cdot x + \beta_0}}$ et $p(x \mid \ominus) = \frac{1}{1 + e^{\beta' \cdot x + \beta_0}}$

Estimation des paramètres $\beta^+ = (\beta, \beta_0)$

En sommant sur toute la base la log-vraisemblance,

$$LL(\beta^+) = \sum_{i=1}^{N} \left[y_i \cdot \left(\beta^{+'} \cdot x_i^+ \right) - \log(1 + \beta^{+'} \cdot x_i^+) \right]$$

On veut maximiser la log-vraisemblance.

$$\frac{\partial LL(\beta^+)}{\partial \beta_i^+} = \sum_{i=1}^{N} x_i \cdot (y_i - p(x_i; \beta^+)$$

Pas de forme simple, il faut utiliser une méthode approchée (Newton-Raphson) utilisant la dérivée seconde (le Hessien) $\frac{\partial^2 LL(\beta^+)}{\partial \beta^+ \partial \beta^{+'}}$.
La mise à jour (jusque convergence) de β^+ prend la forme :

$$\beta_{t+1}^+ = \beta_t^+ - \left(\frac{\partial^2 LL(\beta^+)}{\partial \beta^+ \partial \beta^{+'}} \right)^{-1} \cdot \frac{\partial LL(\beta^+)}{\partial \beta^+}$$

Méthode de Newton (1/2)

fonction de classe C^2

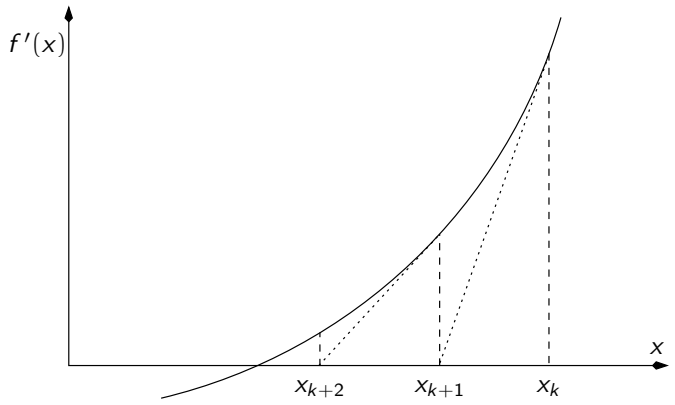
- $f : [a, b] \subset \mathbb{R} \mapsto \mathbb{R}$
- f : 2 fois dérivable
- f'' continue

Méthode de Newton-Raphson : recherche de 0 de la dérivée

- *principe* : engendrer une suite de points (x^k) tendant vers un point stationnaire
- point stationnaire : $f'(x^*) = 0$
- itération k : f' est remplacée par sa linéarisée en x^k : $l(x) = f'(x^k) + [x - x^k]f''(x^k)$
- x^{k+1} déterminé par $l(x^{k+1}) = 0$:

$$\Longrightarrow x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)}$$

Méthode de Newton (2/2)



Un exemple (1/3)

Example

Diabetes data set

- Input X is two dimensional. X_1 and X_2 are the two principal components of the original 8 variables.
- Class 1: without diabetes; Class 2: with diabetes.
- Applying logistic regression, we obtain

$$\beta = (0.7679, -0.6816, -0.3664)^T .$$

From Jia Li (Pensylvania State University)

Un exemple (2/3)

- The posterior probabilities are:

$$Pr(G = 1 \mid X = x) = \frac{e^{0.7679-0.6816X_1-0.3664X_2}}{1 + e^{0.7679-0.6816X_1-0.3664X_2}}$$

$$Pr(G = 2 \mid X = x) = \frac{1}{1 + e^{0.7679-0.6816X_1-0.3664X_2}}$$

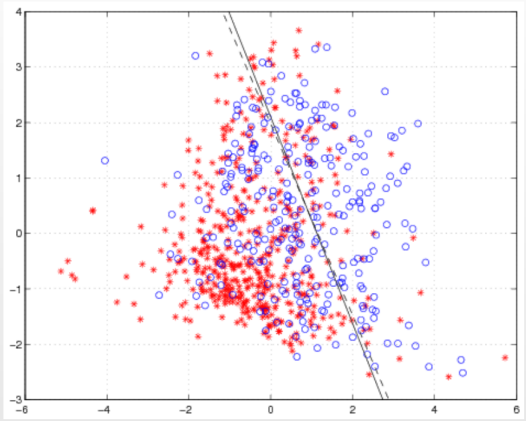
- The classification rule is:

$$\hat{G}(x) = \begin{cases} 1 & 0.7679 - 0.6816X_1 - 0.3664X_2 \geq 0 \\ 2 & 0.7679 - 0.6816X_1 - 0.3664X_2 < 0 \end{cases}$$

From Jia Li (Pensylvania State University)

Un exemple (3/3)

Solid line: decision boundary obtained by logistic regression. Dash line: decision boundary obtained by LDA.



- Within training data set classification error rate: 28.12%.
- Sensitivity: 45.9%.
- Specificity: 85.8%.

From Jia Li (Pensylvania State University)