ÉCOLE CENTRALE DE LILLE

SCIENCE DES DONNEES ET INTELLIGENCE ARTIFICIELLE

PIERRE CHAINAIS

SEMESTRE D'AUTOMNE

Décision et apprentissage

Table des matières

1. Introduction

1.1. Objectifs de l'apprentissage

L'apprentissage comprend :

- La classification supervisée, semi-supervisée et non supervisée : la prédiction est discrète $t \in \{0,1\}$
- La régression : la prédiction est continue $t \in \mathbb{R}$

1.2. Position du problème

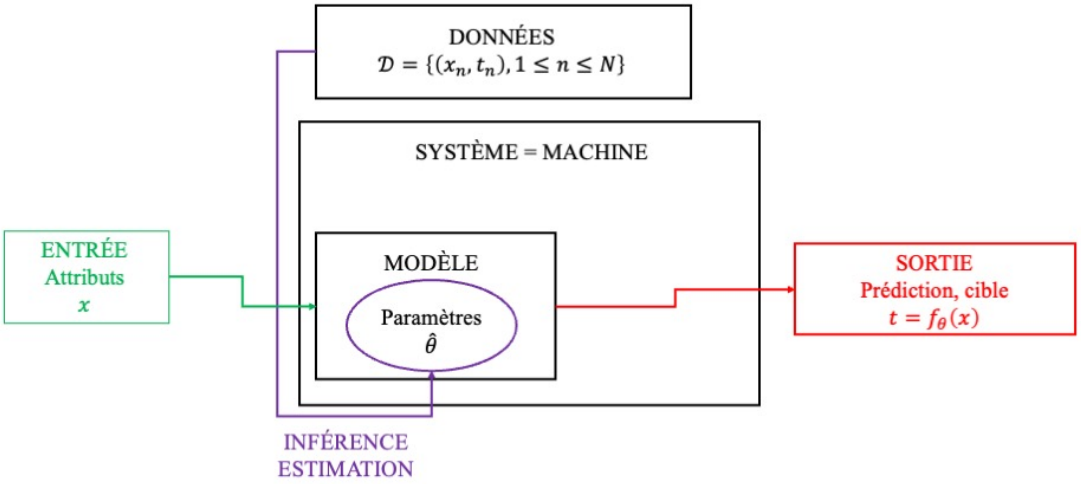


Schéma d'un modèle

Les bonnes propriétés d'un modèle sont la capacité de généralisation et la possibilité de réutilisation du modèle dans un contexte différent.

1.3. Familles d'approches

Dans tous les cas l'objectif est la minimisation des erreurs de prédiction.

Cela peut être la **minimisation d'une fonction de coût** (risque) (par exemple, $\sum|y(x_n) - t_n|^2$). On parle **d'optimisation**.

Cela peut également être la **maximisation d'une probabilité** (par exemple, $p(\theta|\mathcal{D})$). On parle dans ce cas de **raisonnement Bayésien**.

2. Modèles linéaires pour la régression

2.1. Position du problème

Les données sont de la forme $x = (x_1, \dots, x_D) \in \mathbb{R}^D$ et les cibles $t \in \mathbb{R}$.

On peut faire l'hypothèse d'une dépendance linéaire en x et en w : $y(x, w) = w_0 + w_1.x_1 + \cdots + w_D.x_D$.

On peut enrichir le modèle en supprimant la linéarité en x : $y(x, w) = w_1.\phi_0(x) + \cdots + w_M.\phi_M(x) = \langle w, \phi(x) \rangle$ (ϕ traduit le changement de représentation du problème).

2.2. Modèle linéaire en w et en x

Les données sont $\mathcal{D} = \{(x_n, t_n), 1 \leq n \leq N\}$ et le modèle de prédiction est :

$$\boldsymbol{y(x, w) = w_0 + w_1.x_1 + \cdots + w_D.x_D = {}^t\tilde{w}.\tilde{x}}$$

Où :

$${}^t\tilde{x} = (1, x_1, \dots, x_D)$$

On pose :

$$\tilde{X} = \begin{pmatrix} 1 & x_{11} & \dots & x_{1D} \\ \vdots & & & \vdots \\ 1 & x_{N1} & \dots & x_{ND} \end{pmatrix}$$

L'objectif est de minimiser l'erreur quadratique :

$$E_{LS}(\tilde{w}) = \sum_{n=1}^{N} |t_n - y(x_n, \tilde{w})|^2 = \sum_{n=1}^{N} |t_n - {}^t\tilde{w}.\widetilde{x_n}|^2 = \left\| t - \tilde{X}.\tilde{w} \right\|_2^2 = {}^t\left(t - \tilde{X}.\tilde{w}\right).\left(t - \tilde{X}.\tilde{w}\right)$$

$$\tilde{w} = \underset{\tilde{w}}{\operatorname{argmin}}\, E_{LS}(\tilde{w})$$

L'erreur quadratique est convexe et différentiable donc il existe une solution unique. On cherche un optimum en calculant un point on le gradient s'annule.

$$\nabla E_{LS}(\tilde{w}) = \frac{\partial E_{LS}}{\partial \tilde{w}}(\tilde{w}) = -2.\tilde{X}^{\mathrm{T}}.\left(t - \tilde{X}.\tilde{w}\right)$$

$$\Delta E_{LS}(\tilde{w}) = \frac{\partial^2 E_{LS}}{\partial \tilde{w}.\partial {}^t\tilde{w}}(\tilde{w}) = 2.\tilde{X}^{\mathrm{T}}.\tilde{X} \in \mathcal{S}_{D+1}^{+}(\mathbb{R})$$

La matrice est symétrique **définie** positive, i.e. ses valeurs propres sont **strictement** positives si ${}^t\tilde{X}.\tilde{X}$ est de rang plein. Dans ce cas, on a bien la convexité.

Sous cette hypothèse, $\tilde{w}$ est solution unique de :

$$\boldsymbol{\tilde{X}^{\mathrm{T}}.t - \tilde{X}^{\mathrm{T}}.\tilde{X}.\tilde{w} = 0 \Leftrightarrow \tilde{X}^{\mathrm{T}}.t = \tilde{X}^{\mathrm{T}}.\tilde{X}.\tilde{w} \Leftrightarrow \tilde{w} = \left(\tilde{X}^{\mathrm{T}}.\tilde{X}\right)^{-1}.{}^{t}\tilde{X}.t}$$

On appelle pseudo-inverse de Moore-Penrose de $\tilde{X}$ la solution :

$$\boldsymbol{\tilde{X}^{\dagger} = \left(\tilde{X}^{\mathrm{T}}.\tilde{X}\right)^{-1}.{}^{t}\tilde{X}}$$

Si $\tilde{X}$ est inversible, alors $\tilde{X}^{\dagger} = \tilde{X}^{-1}$, mais $\tilde{X}$ est rectangulaire donc cette hypothèse est rarement vérifiée.

Au lieu de résoudre $t = \tilde{X}.\tilde{w}$*, on cherche un minimum de* $\left\|t - \tilde{X}.\tilde{w}\right\|_2^2$*. De plus, la résolution de* $\nabla E_{LS}(\tilde{w}) = 0$ *n'est pas toujours évidente, donc on peut avoir recours à la descente de gradient. Cela a aussi l'avantage de ne pas avoir à inverser de matrice, et donc d'économiser du coût calculatoire.*

Les prédictions sur l'ensemble d'entrainement sont : $t = \tilde{X}.\tilde{w} = \tilde{X}.\left(\tilde{X}^{T}.\tilde{X}\right)^{-1}.\tilde{X}^{T}.\tilde{w}$*. On appelle la matrice* $\tilde{X}.\left(\tilde{X}^{T}.\tilde{X}\right)^{-1}.{}^{t}\tilde{X}$ *« hat matrix ».*

2.3. Modèle linéaire en w et non en x

On pose $\phi_0(\vec{x}) = 1$ et les $\phi_j(x)$ pour $1 \leq i \leq M-1$ (pas forcément linéaires en x).

$$\tilde{\Phi}(X) = \begin{pmatrix} \phi_0(\overrightarrow{x_1}) = 1 & \phi_1(\overrightarrow{x_1}) & \dots & \phi_{M-1}(\overrightarrow{x_1}) \\ \vdots & & & \vdots \\ \phi_0(\overrightarrow{x_N}) = 1 & \phi_1(\overrightarrow{x_N}) & \dots & \phi_{M-1}(\overrightarrow{x_N}) \end{pmatrix}$$

La fonction $\phi: \mathbb{R}^D \to \mathbb{R}^M$ et $\phi(\vec{x}) = \left(\phi_0(\vec{x}), \dots, \phi_{M-1}(\vec{x})\right)$ permet d'obtenir la matrice $\tilde{\Phi}(X)$ de dimension $N \times M$.

Par exemple, si $\vec{x} = (x_1, x_2)$ *et* $\phi(\vec{x}) = (1, x_1^2, x_1.x_2, x_2^2)$*, alors* $M = 4$ *et* $y(\vec{x}, \tilde{w}) = w_0 + w_1.x_1^2 + w_2.x_1.x_2 + w_3.x_2^2$.

$$\tilde{w} = \underset{\tilde{w}}{\operatorname{argmin}}\, {}^{t}\left(t - \tilde{\Phi}(X).\tilde{w}\right).\left(t - \tilde{\Phi}(X).\tilde{w}\right)$$

Donc de manière similaire :

$$\tilde{w} = \tilde{\Phi}^{\dagger}.t \in \mathbb{R}^M$$

Par exemple, pour une régression polynomiale à l'ordre 3 d'un vecteur $\vec{x}$ *de dimension 2 :*

$$\phi(\vec{x}) = (1, x_1, x_2, x_1^2, x_1.x_2, x_2^2, x_1^3, x_1^2.x_2, x_1.x_2^2, x_2^3)$$

Pour des sorties multiples, c'est-à-dire $t \in \mathbb{R}^K$, avec des données sont $\mathcal{D} = \{(x_n, t_n) \in \mathbb{R}^{D \times K}, 1 \leq n \leq N\}$ et le modèle $y\left(\vec{x}, \widetilde{W}\right) = \left(y_k\left(\vec{x}, \widetilde{\overrightarrow{w_k}}\right)\right)$, on a :

$$\widetilde{W} = (\widetilde{w_1}, \dots, \widetilde{w_K}), \qquad y\left(\vec{x}, \widetilde{W}\right) = {}^{t}W.\Phi(\vec{x}) = \left({}^{t}w_k.\Phi(\vec{x})\right)_k$$

On minimise le coût quadratique :

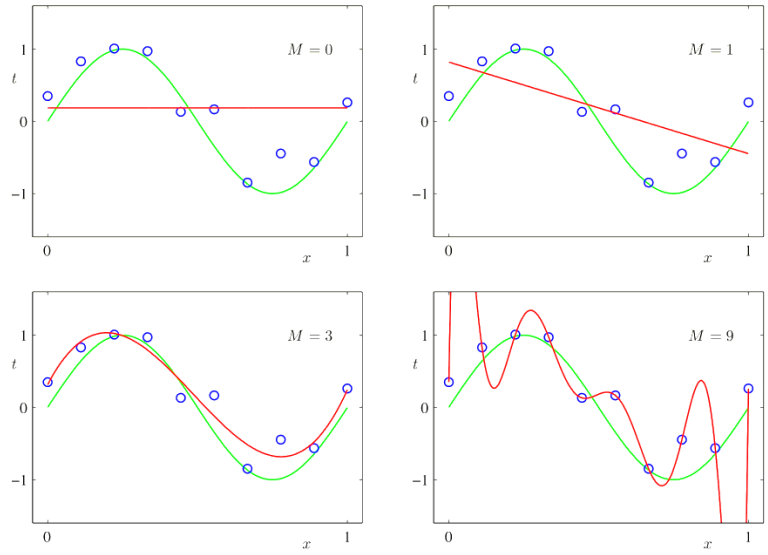
$$\widetilde{W}_{LS} = \left(\widetilde{\Phi^T}.\widetilde{\Phi}\right)^{-1}.\widetilde{\Phi^T}.T \in \mathbb{R}^{M\times K}, \qquad T = (t_{n1}, \dots, t_{nK}) \in \mathbb{R}^{N\times K}$$

2.4. Moindres carrés régularisés et Ridge Regression

La dimension D peut être grande, le nombre d'échantillons N est fini donc limité, le modèle implique M paramètres ($D + 1$ dans le cas où on reste dans la dimension initiale).

L'objectif est d'avoir un modèle précis donc assez riche (pas d'underfitting) mais qui ne surapprend pas (overfitting).

L'idée est de partir d'un modèle potentiellement trop riche et le régulariser, c'est-à-dire, sélectionner un sous-modèle.



Importance de l'ordre sur un modèle polynomial.

2.4.1. La méthode de Ridge Regression

On ne veut pas changer la moyenne mais étudier uniquement les variations autour d'elle. Donc on traite à part le terme d'ordre 0 qu'on fixe comme la moyenne des sorties : $w_0 = \langle t \rangle$; et on ne considère plus les vecteurs $\tilde{x}$ mais x simplement).

On cherche $w = (w_1, \dots, w_{M-1})$ qui minimise :

$$\boldsymbol{E_{RR}(w) = E_{LS}(w) + \lambda.\|w\|_2^2 = \|t - \Phi.w\|_2^2 + \lambda.\|w\|_2^2}$$

Le principe est de minimiser $E_{LS}(w)$ si $\lambda = 0$ et si $\lambda \to +\infty$ de minimiser $\lambda.\|w\|_2^2$. On cherche donc un hyperparamètre λ intermédiaire appelé aussi **paramètre de régularisation**.

Le λ optimal correspond au modèle qui a la meilleure généralisation. Comme λ donne le même poids à tous les w_j, cela nécessite de **normaliser** les données auparavant, c'est-à-dire de le rendre centrés-réduits :

$$\vec{x} \mapsto \overrightarrow{x^{(c)}} = \left(\frac{x_j - \mu}{\sigma}\right)_j, \qquad \mu = \langle x_k \rangle_k, \qquad \sigma = \sqrt{var(x_k)_k}$$

$$\vec{\phi}(\vec{x}) \mapsto \overrightarrow{\phi^{(c)}} = \left(\frac{\phi_j - \mu}{\sigma}\right)_j, \qquad \mu = \langle \phi_k \rangle_k, \qquad \sigma = \sqrt{var(\phi_k)_k}$$

On résout le problème pour $w^{(c)}$ sachant les données centrées-réduites :

$$\mathcal{D}^{(c)} = \left\{\left(x_n^{(c)}, t_n^{(c)}\right) \in \mathbb{R}^{D\times K}, 1 \leq n \leq N\right\}$$

On a :

$$E_{RR}(w) = E_{LS}(w) + \lambda. \|w\|_2^2 = E_{LS}(w) + \lambda. w^T. w$$

Donc :

$$\frac{\partial E_{RR}}{\partial \widetilde{w}}(w) = \frac{\partial E_{LS}}{\partial w}(w) + \lambda. \frac{\partial (w^T. w)}{\partial \widetilde{w}} = -2. \Phi^T. (t - \Phi. w) + 2. \lambda. I_d. w$$

On cherche à résoudre par rapport à w :

$$0 = -2. \Phi^T. (t - \Phi. w) + 2. \lambda. I_d. w$$

La solution est

$$\boldsymbol{w_{RR} = (\Phi^T. \Phi + \lambda. I_d)^{-1}. \Phi^T. t}$$

Donc $w_{RR} = w_{LS}$ si et seulement si $\lambda = 0$. Et w_{RR} est toujours bien définie car $(\Phi^T. \Phi + \lambda. I_d)$ est toujours inversible. Numériquement, il arrive que $\Phi^T. \Phi$ soit mal conditionnée :

$$cond(\Phi^T. \Phi) = \frac{\lambda_{max}}{\lambda_{min}} \gg 1$$

Plus généralement, on peut utiliser d'autres régularisations, par exemple en $\lambda \sum |w_j|^q, q > 0$.

2.4.2. Au-delà de la Ridge Regression

On suppose qu'on observe $t = \Phi. \beta + \varepsilon$, et qu'on a $\Phi^*. \Phi = \mathrm{I_d}$ (base orthonormée) donc Φ^* est le conjugué du transposé. On veut résoudre :

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}\left(\|t - \Phi. \beta\|_2^2 + \lambda. G(\beta)\right)$$

Où $G(\beta) = \sum_{j=1}^{M-1} |\beta_j|^q$, $\beta^* = \Phi^*. \mathrm{t}$.

Si $q = 2$, on parle de Ridge Regression :

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|\beta^* - \beta\|_2^2 + \lambda. \|\beta\|_2^2) = \underset{\beta}{\operatorname{argmin}} \left(\sum_{j=1}^{M-1} |\beta_j^* - \beta_j|^2 + \lambda. |\beta_j|^2\right)$$

$$= \left(\underset{\beta_j}{\operatorname{argmin}} \left(|\beta_j^* - \beta_j|^2 + \lambda. |\beta_j|^2\right)\right)_{1\leq j<M} = \left(\frac{\beta_j^*}{1+\lambda}\right)_{1\leq j<M}$$

Si $q = 0$, on parle de parcimonie :

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\mathbf{argmin}}\left(\|\boldsymbol{\beta}^* - \boldsymbol{\beta}\|_2^2 + \boldsymbol{\lambda}.\#\{\boldsymbol{\beta_j} \neq \mathbf{0}\}\right) = \left(\frac{\boldsymbol{\beta_j^*}}{|\boldsymbol{\beta_j^*}|}.\mathbf{thres}_{\boldsymbol{\lambda/2}}(|\boldsymbol{\beta_j^*}|)\right)_{\mathbf{1 \leq j < M}}$$

On parle de seuillage dur : être supérieur ou inférieur à $\lambda/2$:

$$\text{thres}_{\lambda/2} : x \mapsto \begin{cases} 0 \; si \; |x| \leq \lambda/2 \\ x \; sinon \end{cases}$$

Si $q = 1$, on parle de LASSO :

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\mathbf{argmin}}\left(\sum_{\boldsymbol{j=1}}^{\boldsymbol{M-1}} |\boldsymbol{\beta_j^*} - \boldsymbol{\beta_j}|^2 + \boldsymbol{\lambda}.|\boldsymbol{\beta_j}|\right) = \left(\underset{\boldsymbol{\beta_j}}{\mathbf{argmin}}\left(|\boldsymbol{\beta_j^*} - \boldsymbol{\beta_j}|^2 + \boldsymbol{\lambda}.|\boldsymbol{\beta_j}|\right)\right)_{\mathbf{1 \leq j < M}}$$
$$= \left(\frac{\boldsymbol{\beta_j^*}}{|\boldsymbol{\beta_j^*}|}.\mathbf{soft\ thres}_{\boldsymbol{\lambda/2}}(|\boldsymbol{\beta_j^*}|)\right)_{\mathbf{1 \leq j < M}}$$

On parle de seuillage doux, c'est-à-dire qu'on applique l'opérateur continu :

$$\text{soft thres}_{\lambda/2} : x \mapsto \begin{cases} 0 \; si \; |x| \leq \lambda/2 \\ \dfrac{x}{|x|}.\left(|x| - \dfrac{\lambda}{2}\right) \end{cases}$$

Si la solution est vraiment parcimonieuse, alors la solution LASSO est aussi parcimonieuse.

2.5. Formulation bayésienne du problème de régression linéaire

On adopte une vision probabiliste, c'est-à-dire que le monde est fait de variables aléatoires dont on observe les réalisations :

$$t = y(x, w) + \varepsilon, \qquad \varepsilon \rightsquigarrow \mathcal{N}(0, \beta^{-1}.I)$$

Où β est la précision et β^{-1} la variance, le bruit est indépendant de y. Donc chaque observation $t_n \rightsquigarrow \mathcal{N}(y(x_n, w), \beta^{-1})$.

On cherche les paramètres w, β qui expliquent le plus probablement $p(t_n|x_n, w, \beta)$ de façon **vraisemblable**.

On suppose que les observations sont indépendantes et identiquement distribuées.

On appelle **vraisemblance** :

$$\boldsymbol{\mathcal{L}(w, \beta) = p(t_{1:N}|x_{1:N}, w, \beta) = \prod_{n=1}^{N} p(t_n|x_n, w, \beta) = \left(\frac{\beta}{2.\pi}\right)^{\frac{N}{2}}.e^{-\frac{\beta}{2}.\sum_{n=1}^{N}|y(x_n,w)-t_n|^2}}$$

On appelle **Maximum Likelihood Estimate (MLE) de w** la grandeur :

$$\widehat{w}_{ML} = \underset{w}{\operatorname{argmin}}\big(-\log\big(\mathcal{L}(w,\beta)\big)\big) = \underset{w}{\operatorname{argmin}}\left(\sum_{n=1}^{N}|y(x_n,w)-t_n|^2\right) = \widehat{w}_{LS}$$

On appelle **Maximum Likelihood Estimate (MLE) de $\boldsymbol{\beta^{-1}}$** (variance) la grandeur :

$$\widehat{\beta^{-1}}_{ML} = \underset{\beta^{-1}}{\operatorname{argmin}}\big(-\log\big(\mathcal{L}(\beta)\big)\big) \propto \underset{\beta^{-1}}{\operatorname{argmin}}\left(\frac{N}{2}.\log(\beta)+\frac{\beta}{2}.\sum_{n=1}^{N}|y(x_n,w)-t_n|^2\right)$$

On cherche donc les arguments d'annulation de :

$$-\frac{\partial\big(\log\big(\mathcal{L}(\beta)\big)\big)}{\partial\beta} = \frac{N}{2\beta}+\frac{1}{2}.\sum_{n=1}^{N}|y(x_n,w)-t_n|^2$$

C'est-à-dire :

$$\widehat{\beta^{-1}}_{ML} = \frac{1}{N}.\sum_{n=1}^{N}|y(x_n,w)-t_n|^2$$

Avec régularisation (donc avec un a priori sur w), on ajoute par exemple $\lambda.\|w\|_2^2$. On peut prendre le chemin inverse en appliquant l'opérateur $x \mapsto e^{-x}$ et on obtient $e^{-\frac{\alpha}{2}.\|w\|_2^2}$, une loi gaussienne de précision α et de moyenne 0.

$$p(w|x_{1:N},t_{1:N},\beta,\alpha) \propto p(t_{1:N}|x_{1:N},w,\beta,\alpha).p(w|\alpha)$$

On appelle **loi a posteriori** $p(w|x_{1:N},t_{1:N},\beta,\alpha)$, la **loi a priori** $p(w|\alpha)$ et la **vraisemblance** $p(t_{1:N}|x_{1:N},w,\beta,\alpha)$ et :

$$posterior \propto likelihood \times prior$$

On appelle **Maximum A Posteriori (MAP) de $\boldsymbol{w}$** la grandeur :

$$\begin{aligned}\widehat{\boldsymbol{w}}_{\boldsymbol{MAP}} &= \underset{\boldsymbol{w}}{\mathbf{argmax}}\big(\boldsymbol{p}(\boldsymbol{w})\big)\\ &= \underset{\boldsymbol{w}}{\mathbf{argmax}}\left(\boldsymbol{\mathcal{L}}(\boldsymbol{w},\boldsymbol{\beta}).\boldsymbol{e}^{-\frac{\boldsymbol{\alpha}}{\boldsymbol{2}}.\|\boldsymbol{w}\|_{\boldsymbol{2}}^{\boldsymbol{2}}}\right) = \underset{\boldsymbol{w}}{\mathbf{argmax}}\left(\mathbf{log}\left(\boldsymbol{\mathcal{L}}(\boldsymbol{w},\boldsymbol{\beta}).\boldsymbol{e}^{-\frac{\boldsymbol{\alpha}}{\boldsymbol{2}}.\|\boldsymbol{w}\|_{\boldsymbol{2}}^{\boldsymbol{2}}}\right)\right)\\ &= \underset{\boldsymbol{w}}{\mathbf{argmin}}\left(\frac{\boldsymbol{\beta}}{\boldsymbol{2}}.\sum_{\boldsymbol{n=1}}^{\boldsymbol{N}}|\boldsymbol{y}(\boldsymbol{x_n},\boldsymbol{w})-\boldsymbol{t_n}|^{\boldsymbol{2}}+\frac{\boldsymbol{\alpha}}{\boldsymbol{2}}.\|\boldsymbol{w}\|_{\boldsymbol{2}}^{\boldsymbol{2}}\right) = \underset{\boldsymbol{w}}{\mathbf{argmin}}\big(\boldsymbol{E_{RR}}(\boldsymbol{w})\big)\\ &= \widehat{\boldsymbol{w}}_{\boldsymbol{RR}}\end{aligned}$$

Pour régler α et β, on utilise $\hat{\beta}_{MAP} = \hat{\beta}_{ML}$ et on règle α comme un hyperparamètre. On utilise la cross-validation par exemple pour estimer $E_{RR}(w|\alpha)$, qui nous sert ensuite à minimiser α.

Cette approche permet d'évaluer les incertitudes grâce aux formules de probabilité de w, ce qui n'était pas possible avec les méthodes de régression.

2.6. Fonction de coût et estimateur théorique

On a la fonction de coût $L(y(x), t)$ et on cherche à estimer la fonction y idéalement en minimisant :

$$E(L) = \iint L(y(x), t)p(x, t).dx.dt$$

On l'écrit dans le cas $L(y(x), t) = (y(x) - t)^2$:

$$E(L) = \iint (y(x) - t)^2.p(x, t).dx.dt$$

Donc on cherche :

$$\frac{\partial E(L)}{\partial y} = 2.\int (y(x) - t).p(x, t).dt = 0$$

Donc :

$$\int y(x).p(x, t).dt = \int t.p(x, t).dt$$

Donc :

$$y(x).p(x) = \int t.p(t|x).p(x).dt$$

On obtient :

$$y(x) = \int t.p(t|x).dt = E(t|x)$$

On s'intéresse maintenant à la fonction de coût de L^0 où les L^q sont définies par :

$$L^q(y(x), t) = |y(x) - t|^q$$

Lorsque $q \to 0$, $y(x) = \text{mode}(t|x)$ et on a vu que pour $q = 1$, $y(x) = \text{median}(t|x)$.

2.7. Compromis biais variance

On cherche un modèle y qui approche h. On fait l'hypothèse sur la génération des données, avec h le système réel et ε un bruit indépendant de h, d'espérance nulle et de variance finie :

$$t = h(x) + \varepsilon$$

Le biais b se trouve entre $h(x)$ et $E_{\mathcal{D}}\big(y(x, \mathcal{D})\big)$ et la variance entre cette espérance et $y(x, \mathcal{D})$.

Les trois sources d'écart sont donc la variance du bruit, la variance liée aux données et le biais de la modélisation.

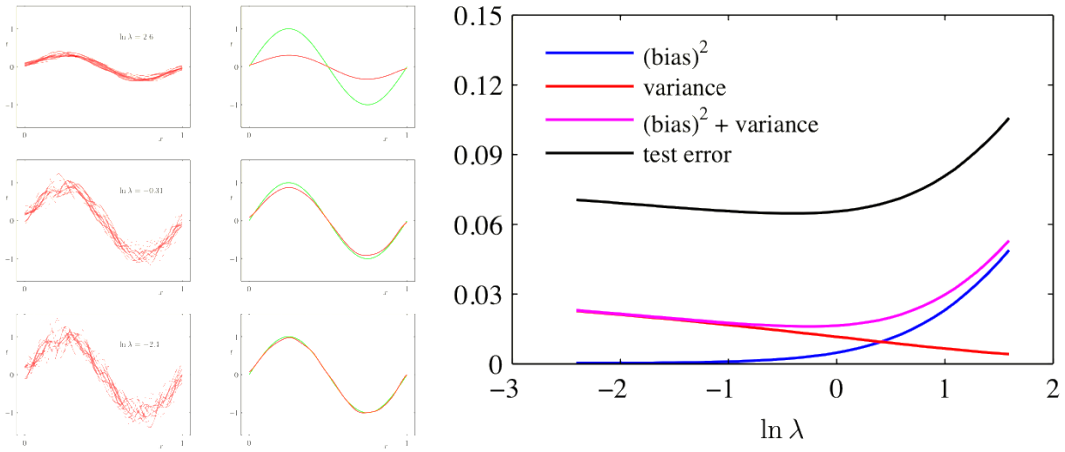
Alors :

$$E_{\mathcal{D}}(|y(x,\mathcal{D}) - t|^2) = E_{\mathcal{D}}(|y(x,\mathcal{D}) - h(x) + h(x) - t|^2)$$
$$= E_{\mathcal{D}}(|y(x,\mathcal{D}) - h(x) + h(x) - t|^2)$$
$$= E_{\mathcal{D}}(|y(x,\mathcal{D}) - h(x)|^2) + E_{\mathcal{D}}(|h(x) - t|^2)$$
$$+ 2.E_{\mathcal{D}}\left(\left|\left(y(x,\mathcal{D}) - h(x)\right).(h(x) - t)\right|^2\right)$$

Donc :

$$E_{\mathcal{D}}(|y(x,\mathcal{D}) - t|^2) = b^2 + \mathrm{var}_{\mathcal{D}}\big(y(x,\mathcal{D})\big) + \sigma_\varepsilon^2$$

Et σ_ε^2 est lié à l'erreur sur la qualité des mesures (incompressible), b^2 à l'adéquation du modèle au système réel et $\mathrm{var}_{\mathcal{D}}\big(y(x,\mathcal{D})\big)$ à la sensibilité aux données.



Compromis biais variance.

3. Classification et théorie de la décision

3.1. Définitions

Les données d'entrainements sont de la forme $\mathcal{D} = \{(x_n, t_n), 1 \leq n \leq N\}$ où $x_n \in \mathcal{X} \subset \mathbb{R}^D$ et $t_n \in \mathcal{T}$ avec $\text{card}(\mathcal{T}) = K$.

Typiquement, $t_n = 0$ si $x_n \in \mathcal{C}_1$ et $t_n = 1$ si $x_n \in \mathcal{C}_2$.

Si $K > 2$, $t_n = (0, \dots, 1, \dots, 0)$ avec le 1 en position de la classe de t_n.

3.2. Erreur de prédiction, fonction de coût

La fonction de coût 0-1 :

$$\boldsymbol{L(a, b) = \chi_{\{a \neq b\}}}$$

Le taux d'erreur réelle est :

$$E(f) = E_{t,x}\Big(L\big(t, f(x)\big)\Big)$$

Et devient avec la fonction de coût 0-1 :

$$E(f) = P(f(x) \neq t)$$

Le taux d'erreur empirique :

$$\boldsymbol{E_N(f) = \frac{1}{N} . \sum_{n=1}^{N} L\big(t_n, f(x_n)\big)}$$

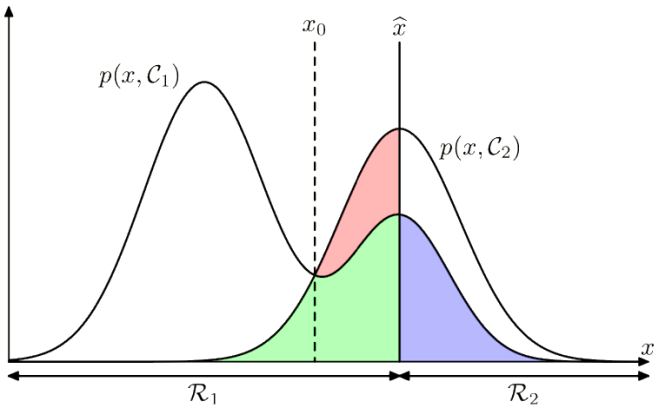
3.3. Classification binaire

Ici, on a $x \in \mathbb{R}, t \in \{0,1\}$ et donc deux régions de décision :

$$\mathcal{R}_1 = \{x, f(x) = 0\}, \qquad \mathcal{R}_2 = \{x, f(x) = 2\}$$

Alors, l'espérance de la fonction de coût que l'on veut minimiser :

$$E(f) = P(x \in \mathcal{R}_1, t = 1) + P(x \in \mathcal{R}_2, t = 0) = \int_{\mathcal{R}_1} p(x, t = 1).\, dx + \int_{\mathcal{R}_2} p(x, t = 0).\, dx$$



Fonctions de densité des régions 1 et 2.

Ici, $\hat{x}$ est le seuil de décision et l'erreur est la somme des aires colorées. L'optimum est l'endroit où la zone rouge disparaît, i.e. $\hat{x} = x_0$. En plaçant ce seuil, on détermine les régions $\mathcal{R}_1$ et $\mathcal{R}_2$ de telle sorte que :

$$\forall x \in \mathcal{R}_1, \quad p(x, t=1) \leq p(x, t=0), \quad \forall x \in \mathcal{R}_2, \quad p(x, t=0) \leq p(x, t=1)$$

3.4. Décision statistique théorique

La **règle de Bayes** donne le **maximum a posteriori (MAP)** :

$$f^*(x) = \underset{k}{\operatorname{argmax}}\big(p(x, \mathcal{C}_k)\big) = \underset{k}{\operatorname{argmax}}\big(P(\mathcal{C}_k|x)\big)$$

On appelle **Taux d'erreur Bayésien** la quantité $E(f^*)$.

La règle de bayes pour la classification est optimale, i.e. pour toute autre règle f, $E(f^*) \leq E(f)$.

3.5. Inférence et décision : familles de modèles

L'inférence est la détermination de $p(\mathcal{C}_k|x)$ et la **décision** est d'utiliser cette quantité pour affecter des classes.

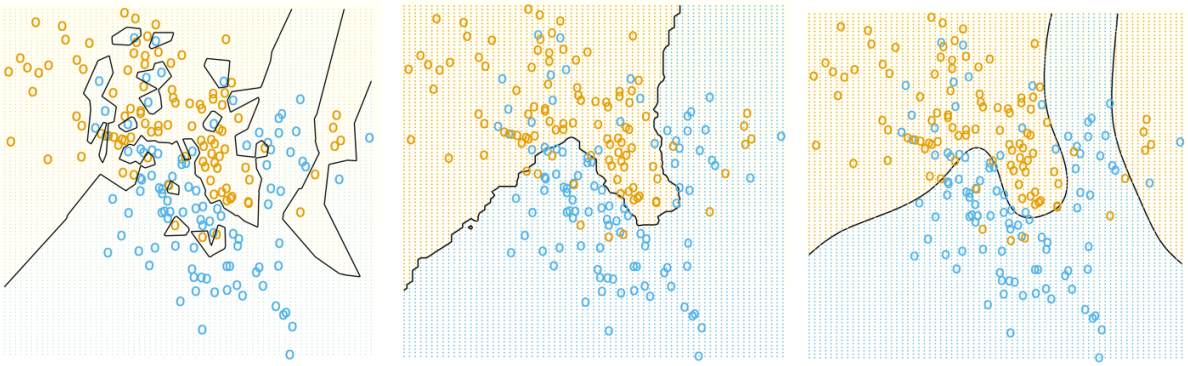
Un modèle génératif utilise $p(x|\mathcal{C}_k)$et $p(\mathcal{C}_k)$ pour déduire $p(\mathcal{C}_k|x)$. L'interprétation de ce modèle est simple mais ce dernier peut être mal adapté (discriminant linear analysis par exemple).

Un modèle discriminant estime directement $p(\mathcal{C}_k|x)$ (régression logistique par exemple).

Une fonction discriminante estime $f(x)$ directement (k-NN par exemple). Ces modèles peuvent être très efficaces mais sont souvent peu interprétables.

3.6. k-NN : k plus proches voisins

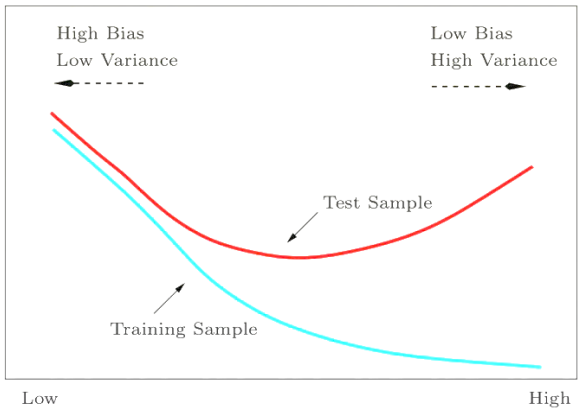
On classifie x dans la classe dont la majorité de ses k plus proches voisins fait partie.



Classification des données pour $k = 1$ et $k = 15$ et règle de bayes.

Lorsque $k = 1$, la fonction est trop riche car elle dépend de N variables.

3.7. Sélection de modèle



Erreur de prédiction en fonction de la complexité du modèle.

4. Modèles linéaires pour la classification supervisée

On cherche toujours à prédire une classe $\mathcal{C}_k$ pour des entrées x. L'idée est de définir k régions $\mathcal{R}_k$ qui sont séparées par des hyperplans en guise de frontières de décision.

4.1. Frontières discriminantes linéaires

Un hyperplan séparateur est de la forme :

$$\mathcal{H} = \{x \in \mathbb{R}^D, y(x) = \tilde{w}^T.\tilde{x} = 0\}$$

Où $y(x) = w^T.x + w_0$ avec w la normale à l'hyperplan et w_0 le biais.

Une règle de décision peut séparer en deux régions $\mathcal{R}_1 = \{x, y(x) < 0\}$ et $\mathcal{R}_2 = \{x, y(x) \geq 0\}$.

On travaille avec K classifieurs pour éviter les zones d'indécision.

On définit K fonctions $y_k(x) = \tilde{w}_k{}^T.\tilde{x}$ et la fonction de classification :

$$f(x) = \underset{k}{\operatorname{argmax}}\, y_k(x)$$

4.2. Moindres carrés pour la classification

L'idée de faire une régression par moindres carrés pour apprendre $\widehat{p(\mathcal{C}_k|x)}$ n'est pas une bonne idée car elle estime une droite de classification et non un segment : on ne cherche pas à prédire un réel mais une probabilité.

De plus les évènements extrêmes exercent une influence excessive à l'entrainement.

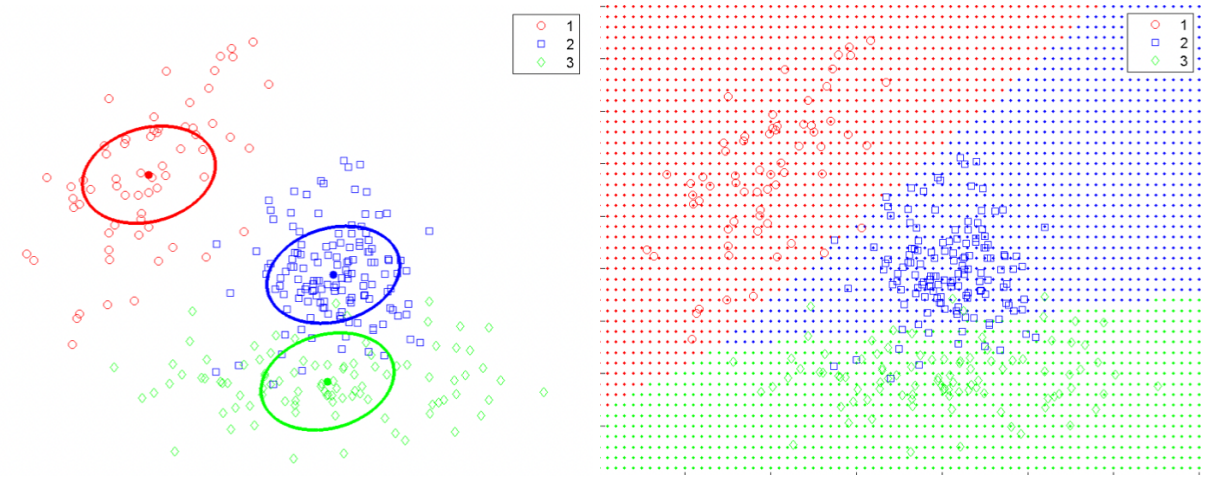
Enfin, il y a un phénomène de masquage lorsqu'une classe est intercalée entre d'autres.

Il ne faut donc pas utiliser les moindres carrés pour la classification.

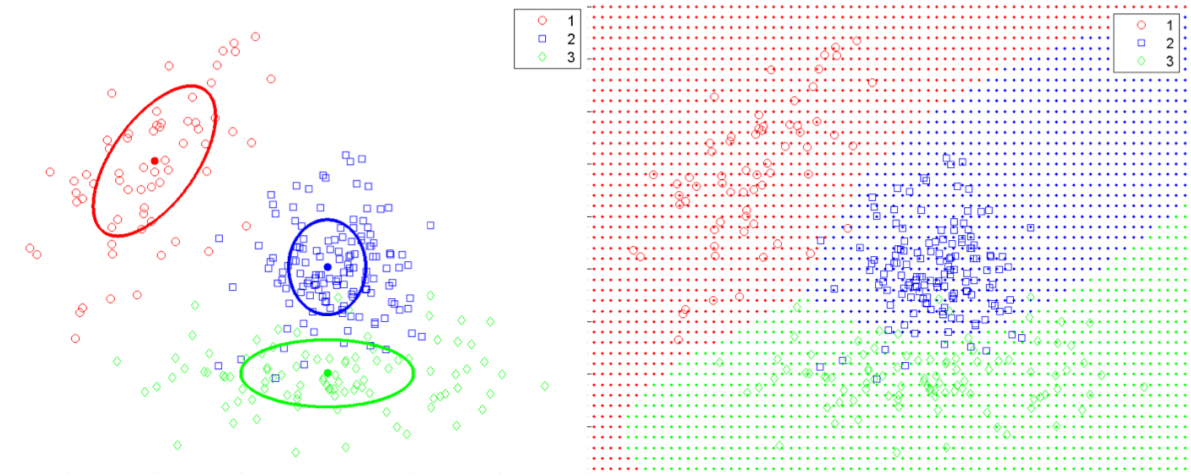
4.3. Analyse discriminante linéaire / quadratique (LDA / QDA)

On modélise les probabilités par des gaussiennes (ellipses) $p(x|\mathcal{C}_k) \rightsquigarrow \mathcal{N}(x|\mu_k, \Sigma_k)$ où μ_k est le centre de l'ellipse et Σ_k l'étalement selon les directions. Les ellipses sont dirigées selon les vecteurs propres et l'étalement est selon les valeurs propres.

L'analyse discriminante linéaire suppose $\Sigma_k = \Sigma$ et définit donc des droites de séparation. L'analyse discriminante quadratique fait intervenir des matrices de covariance différentes et définit donc des courbes.



Classification par LDA.



Classification par QDA.

Comme ces modèles sont génératifs, on calcule :

$$p(x|\mathcal{C}_k) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_k|}.e^{-\frac{1}{2}.(w-\mu_k)^T.\Sigma_k^{-1}.(x-\mu_k)}$$

Et d'autre part, on pose :

$$p(\mathcal{C}_k) = \Pi_k$$

Donc :

$$p(x,\mathcal{C}_k) = p(x|\mathcal{C}_k).p(\mathcal{C}_k) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_k|}.e^{-\frac{1}{2}.(w-\mu_k)^T.\Sigma_k^{-1}.(x-\mu_k)}.\Pi_k$$

On utilise le maximum de vraisemblance pour estimer les paramètres $\mu_{1:K}, \Sigma_{1:K}, \Pi_{1:K}$:

$$p(x_n,t_n|\mu_{1:K},\Sigma_{1:K},\Pi_{1:K}) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_{t_n}|^{\frac{1}{2}}}.e^{-\frac{1}{2}.(x_n-\mu_{t_n})^T.\Sigma_{t_n}^{-1}.(x_n-\mu_{t_n})}.\Pi_{t_n}$$

La vraisemblance (de toutes les observations sachant tous le paramètres) est :

$$\mathcal{L}(x_{1:N}, t_{1:N}|\mu_{1:K}, \Sigma_{1:K}, \Pi_{1:K}) = \prod_{n=1}^{N} p(x_n, t_n|\mu_{1:K}, \Sigma_{1:K}, \Pi_{1:K})$$

La neg-log-vraisemblance est :

$$-\ln\big(\mathcal{L}(x_{1:N}, t_{1:N}|\mu_{1:K}, \Sigma_{1:K}, \Pi_{1:K})\big) = \ln\left(\prod_{n=1}^{N} \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_{t_n}|}.e^{-\frac{1}{2}.(x_n-\mu_{t_n})^T.\Sigma_{t_n}^{-1}.(x_n-\mu_{t_n})}.\Pi_{t_n}\right)$$
$$= \frac{1}{2}.\sum_{n=1}^{N}\ln|\Sigma_{t_n}| + \frac{1}{2}.\sum_{n=1}^{N}(x_n-\mu_{t_n})^T.\Sigma_{t_n}^{-1}.(x_n-\mu_{t_n}) + \sum_{n=1}^{N}\ln\Pi_{t_n} + K$$

Expression des $\boldsymbol{\mu_k}$

Pour trouver l'expression de μ_k, on dérive :

$$\frac{\partial}{\partial\mu_k}(-\ln(\mathcal{L})) = \sum_{\substack{n=1\\t_n=k}}^{N}.\Sigma_k^{-1}.(x_n-\mu_k)$$

On cherche l'optimum :

$$\Sigma_k^{-1}.\left(\sum_{x_n\in\mathcal{C}_k} x_n - \sum_{x_n\in\mathcal{C}_k}\mu_k\right) = 0$$

Or Σ_k est définie positive en posant $N_k = \#\mathcal{C}_k$:

$$\sum_{x_n\in\mathcal{C}_k} x_n - N_k.\mu_k = 0$$

Donc :

$$\hat{\mu}_k = \frac{1}{N_k}.\sum_{x_n\in\mathcal{C}_k} x_n$$

Expression des $\boldsymbol{\Pi_k}$

Pour trouver l'expression de Π_k, on réécrit :

$$-\ln(\mathcal{L}) = \sum_{n\in\mathcal{C}_k}\ln\big(\Pi_k.p_k(x_n|\mu_k,\Sigma_k)\big) + \sum_{n\notin\mathcal{C}_k}\ln\big((1-\Pi_k).p_{-k}(x_n|\mu_{-k},\Sigma_{-k})\big)$$

On dérive :

$$\frac{\partial}{\partial\Pi_k}(-\ln(\mathcal{L})) = \frac{N_k}{\Pi_k} - \frac{N-N_k}{1-\Pi_k}$$

On cherche l'optimum :

$$\frac{N_k}{\Pi_k} - \frac{N - N_k}{1 - \Pi_k} = 0$$

Donc :

$$\widehat{\Pi}_k = \frac{N_k}{N}$$

Expression des $\boldsymbol{\Sigma_k}$

Pour trouver l'expression de Σ_k, on réécrit :

$$-\ln(\mathcal{L}) = \frac{N_k}{2}.\ln|\Sigma_k| + \frac{1}{2}.\sum_{n \in \mathcal{C}_k} \mathrm{Tr}\Big((x_n - \mu_k)^T.\Sigma_k^{-1}.(x_n - \mu_k)\Big) + K$$

On dérive :

$$\frac{\partial}{\partial \Sigma_k}(-\ln(\mathcal{L})) = -\frac{N_k}{2}.\Sigma_k + \frac{1}{2}.\sum_{n \in \mathcal{C}_k}(x_n - \mu_k).(x_n - \mu_k)^T$$

On cherche l'optimum, en posant $S_k = \sum_{n \in \mathcal{C}_k}(x_n - \mu_k).(x_n - \mu_k)^T$:

$$-\frac{N_k}{2}.\Sigma_k + \frac{1}{2}.S_k = 0$$

Donc, on obtient la covariance empirique :

$$\hat{\Sigma}_k = \frac{1}{N_k}.S_k$$

Résumé des expressions

Donc en QDA :

$$\boldsymbol{\widehat{\mu}_k = \frac{1}{N_k}.\sum_{x_n \in \mathcal{C}_k} x_n}, \qquad \boldsymbol{\widehat{\Pi}_k = \frac{N_k}{N}}, \qquad \boldsymbol{\widehat{\Sigma}_k = \frac{1}{N_k}.S_k}$$

Et en LDA :

$$\boldsymbol{\widehat{\Sigma}_k = \widehat{\Sigma} = \frac{1}{N}.\sum_{k=1}^{N} S_k}$$

Frontières de décision

La frontière entre les classes j, k :

$$F_{j,k} = \Big\{x, \qquad \ln\big(p(x|\mu_k, \Sigma).P(\mathcal{C}_k)\big) = \ln\Big(p\big(x\big|\mu_j, \Sigma\big).P\big(\mathcal{C}_j\big)\Big)\Big\}$$

Donc $\forall x \in F_{j,k}$:

$$(x-\mu_k)^T.\Sigma^{-1}.(x-\mu_k) - \left(x-\mu_j\right)^T.\Sigma^{-1}.\left(x-\mu_j\right) + \ln(\Pi_k) - \ln\left(\Pi_j\right) = 0$$

Donc :

$$x^T.\Sigma^{-1}.\left(\mu_k-\mu_j\right) - \ln\left(\frac{\Pi_k}{\Pi_j}\right) + \frac{1}{2}.\left({\mu_k}^T.\Sigma^{-1}.\mu_k - {\mu_j}^T.\Sigma^{-1}.\mu_j\right) = 0$$

On peut réécrire avec w la normale à l'hyperplan :

$$x^T.w - \ln\left(\frac{\Pi_k}{\Pi_j}\right) + w_0 = 0$$

Fonction de décision

La fonction de décision est :

$$\boldsymbol{f(x) = \underset{k}{argmax}\, y_k(x)}$$

Où en QDA :

$$\boldsymbol{y_k(x) = -\frac{1}{2}.(x-\mu_k)^T.\Sigma_k^{-1}.(x-\mu_k) + \ln(\Pi_k) - \frac{1}{2}.\ln|\Sigma_k|}$$

Et en LDA :

$$\boldsymbol{y_k(x) = x^T.\Sigma^{-1}.\mu_k + \ln(\Pi_k) - \frac{1}{2}.{\mu_k}^T.\Sigma^{-1}.\mu_k}$$

4.4. Approche bayésienne naïve

On traite tout d'abord le cas des attributs binaires :

$$x = (x_1, \dots, x_D) \in \{0,1\}^D$$

La probabilité conditionnelle $P(x|\mathcal{C}_k)$ peut prendre 2^D valeurs, nombre qui explose lorsque D augmente.

On fait donc l'hypothèse simplificatrice d'**approche bayésienne naïve** : on suppose que les attributs sont indépendants conditionnellement aux classes :

$$\boldsymbol{P(x|\mathcal{C}_k) = \prod_{i=1}^{D} P(x_i|\mathcal{C}_k)}$$

L'avantage est que la probabilité ne prend plus que D valeurs et :

$$P(x_i|\mathcal{C}_k) = \Pi_{k_i}^{x_i}.\left(1-\Pi_{k_i}\right)^{1-x_i}, \qquad x_i \in \{0,1\}$$

Cette méthode a de bonnes performances en pratique.

Les données sont de la forme $\mathcal{D} = \{(x_n, t_n), 1 \leq n \leq N\}$. Alors :

$$P(x_n, t_n) = P(x_n|t_n).P(t_n) = \left(\prod_{i=1}^{D} P(x_n|t_n)\right).P(t_n) = \left(\prod_{i=1}^{D} \Pi_{t_{n_i}}^{x_i}.\left(1 - \Pi_{t_{n_i}}\right)^{1-x_i}\right).\rho_{t_n}$$

La vraisemblance s'écrit :

$$\mathcal{L} = \prod_{n=1}^{N}\left(\left(\prod_{i=1}^{D} \Pi_{t_{n_i}}^{x_i}.\left(1 - \Pi_{t_{n_i}}\right)^{1-x_i}\right).\rho_{t_n}\right)$$

On cherche des estimateurs $\hat{\rho}_k, \widehat{\Pi}_{k_i}$.

$$\ln \mathcal{L}(\Pi_{k_i}) = \sum_{\substack{n=1 \\ t_n=k}}^{N} x_{n_i}.\ln(\Pi_{k_i}) + (1 - x_{n_i}).\ln(1 - \Pi_{k_i}) + K$$

En posant $n_{k_i} = \#\{x_n \in \mathcal{C}_k, x_{n_i} = 1\}$

$$\frac{\partial}{\partial \Pi_{k_i}}(\ln \mathcal{L}) = \frac{1}{\Pi_{k_i}.(1 - \Pi_{k_i})}.\left(\sum_{\substack{n=1 \\ t_n=k}}^{N} x_{n_i}\right) - \frac{1}{1 - \Pi_{k_i}}.\left(\sum_{\substack{n=1 \\ t_n=k}}^{N} 1\right) = \frac{n_{k_i}}{\Pi_{k_i}.(1 - \Pi_{k_i})} - \frac{N_k}{1 - \Pi_{k_i}}$$

Donc en l'optimum :

$$\frac{n_{k_i}}{\Pi_{k_i}.(1 - \Pi_{k_i})} - \frac{N_k}{1 - \Pi_{k_i}} = 0$$

Donc :

$$\boldsymbol{\widehat{\Pi}_{k_i} = \frac{n_{k_i}}{N_k}}$$

Pour $\hat{\rho}_k$, il faut faire attention au fait que $\sum \rho_k = 1$ de la même manière qu'en LDA et on obtient :

$$\boldsymbol{\widehat{\rho}_k = \frac{N_k}{N}}$$

Donc la fonction de décision est :

$$f(x) = \underset{k}{\operatorname{argmax}}\, \hat{P}(\mathcal{C}_k|x) = \underset{k}{\operatorname{argmax}}\, y_k(x)$$

Où :

$$\hat{P}(\mathcal{C}_k|x) \propto \hat{P}(x|\mathcal{C}_k).\hat{P}(x)$$

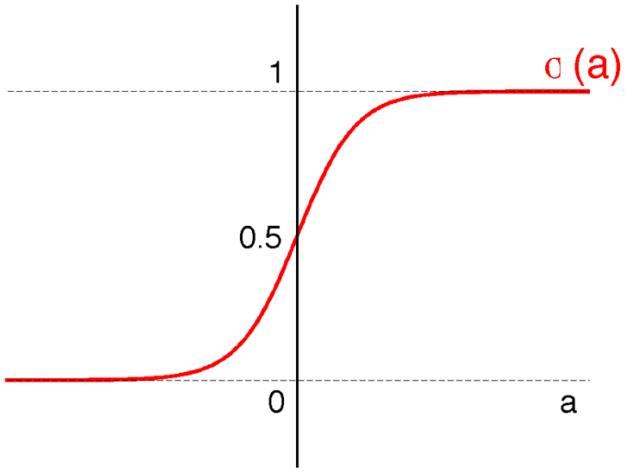
Donc $y_k(x)$ est linéaire en x, donc les séparateurs sont des hyperplans :

$$\boldsymbol{y_k(x) = \ln\left(\widehat{P}(\mathcal{C}_k|x)\right) = \sum_{i=1}^{D} x_i . \ln\left(\frac{n_{k_i}}{N_k}\right) + (1 - x_i) . \ln\left(1 - \frac{n_{k_i}}{N_k}\right) + \ln\left(\frac{N_k}{N}\right)}$$

4.5. Régression logistique

En régression logistique, on écrit une probabilité continue entre 0 et 1 :

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$



Fonction de probabilité en régression logistique.

On modélise les K classes grâce aux $P(\mathcal{C}_k|x)$ avec une dépendance linéaire en $\tilde{w}^T . \tilde{x}$ et $\sum_{k=1}^{K} P(\mathcal{C}_k|x) = 1$.

$$\forall k \neq K, \qquad \frac{\ln\left(P(\mathcal{C}_k|x)\right)}{P(\mathcal{C}_K|x)} = \tilde{w}^T . \tilde{x}$$

La classe $\mathcal{C}_K$ sert de référence.

On montre que :

$$\boldsymbol{\forall k \neq K, \qquad \widehat{P}(\mathcal{C}_k|x) = \frac{e^{\tilde{w}^T . \tilde{x}}}{1 + \sum_{l=1}^{K-1} e^{\tilde{w}^T . \tilde{x}}}, \qquad \widehat{P}(\mathcal{C}_K|x) = \frac{1}{1 + \sum_{l=1}^{K-1} e^{\tilde{w}^T . \tilde{x}}}}$$

Dans le cas binaire, on a :

$$P(x_n, t_n) \propto P(t_n|x_n) = P(\mathcal{C}_1|x_n)^{t_n} . \left(1 - P(\mathcal{C}_1|x_n)\right)^{1-t_n}$$

Donc la vraisemblance :

$$\mathcal{L} \propto \prod_{n=1}^{N} y_n^{t_n} . (1 - y_n)^{1-t_n}$$

Où :

$$y_n = \sigma(\tilde{w}^T.\tilde{x}) = \hat{P}(\mathcal{C}_1|x_n)$$

La fonction de coût :

$$E(\tilde{w}) = -\ln\big(\mathcal{L}(\tilde{w})\big) = -\sum_{n=1}^{N} t_n.\ln(y_n) + (1-t_n).\ln(1-y_n)$$

On a la propriété :

$$\sigma'(a) = \sigma(a).\big(1-\sigma(a)\big)$$

Donc :

$$\frac{\partial}{\partial a}\big(\ln\big(\sigma(a)\big)\big) = 1-\sigma(a), \qquad \frac{\partial}{\partial a}\big(\ln\big(1-\sigma(a)\big)\big) = -\sigma(a)$$

Et on obtient :

$$\nabla_{\tilde{w}}E = \frac{\partial E}{\partial \tilde{w}} = \sum_{n=1}^{N} -t_n.(1-y_n).\tilde{x}_n + (1-t_n).y_n.\tilde{x}_n = \sum_{n=1}^{N}(y_n-t_n).\tilde{x}_n = \tilde{X}^T.(y-t)$$

Où :

$$X \in \mathbb{R}^{D\times N}, \qquad y,t \in \mathbb{R}^N$$

On doit résoudre le système non-linéaire avec une solution numérique (algorithme de Newton-Raphson) :

$$\tilde{X}^T.(\sigma(\tilde{w}^T.\tilde{x}) - t) = 0$$

A chaque itération :

$$\tilde{w}^{new} = \tilde{w}^{old} - H^{-1}.\nabla_{\tilde{w}}E$$

Où :

$$H = \frac{\partial^2 E}{\partial \tilde{w}^T \partial \tilde{w}} = \frac{\partial E}{\partial \tilde{w}^T}\left(\sum_{n=1}^{N} \tilde{x}_n.(\sigma(\tilde{w}^T.\tilde{x}) - t)\right) = \tilde{X}^T.R.\tilde{X}$$

Avec :

$$R = \mathrm{diag}\big(y_n.(1-y_n)\big)_{1\le n\le N}$$