Correction du Partiel de Classification du Mardi 17 janvier 2012.

January 21, 2012

1) Règle de décision Bayésienne (2pts)

La règle de décision Bayésienne associée consiste à accepter l'hypothèse ω_i si

$$P(\omega_i|\mathbf{x}) \geq P(\omega_j|\mathbf{x}), \qquad \forall j = 1,...,k \Longleftrightarrow f(\mathbf{x}|\omega_i) P(\omega_i) \geq f(\mathbf{X}|\omega_j) P(\omega_j) \qquad \forall j = 1,...,k.$$

Puisque toutes les classes sont équiprobables et que les matrices de covariances $\mathbf{\Sigma}_i$ sont toutes égales à $\sigma^2 I_n$, cette régle consiste à affecter x à la classe ω_i (ce que l'on notera $d^*(\mathbf{x}) = \omega_i$) si $\forall j$

$$\frac{p_i}{(2\pi)^{n/2}\sqrt{|\mathbf{\Sigma}_i|}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{m}_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{x}-\mathbf{m}_i)\right] \geq \frac{p_j}{(2\pi)^{n/2}\sqrt{|\mathbf{\Sigma}_j|}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{m}_j)^T \mathbf{\Sigma}_j^{-1}(\mathbf{x}-\mathbf{m}_j)\right]$$

On en déduit

$$\begin{aligned}
d^*(\mathbf{x}) &= \omega_i \Leftrightarrow (\mathbf{x}-\mathbf{m}_i)^T(\mathbf{x}-\mathbf{m}_i) \leq (\mathbf{x}-\mathbf{m}_j)^T(\mathbf{x}-\mathbf{m}_j) \qquad \forall j \\
d^*(\mathbf{x}) &= \omega_i \Leftrightarrow -2\mathbf{m}_i^T\mathbf{x} + \mathbf{m}_i^T\mathbf{m}_i \leq -2\mathbf{m}_j^T\mathbf{x} + \mathbf{m}_j^T\mathbf{m}_j \qquad \forall j \\
d^*(\mathbf{x}) &= \omega_i \Leftrightarrow 2(\mathbf{m}_j-\mathbf{m}_i)^T\mathbf{x} \leq \mathbf{m}_j^T\mathbf{m}_j - \mathbf{m}_i^T\mathbf{m}_i \qquad \forall j \\
d^*(\mathbf{x}) &= \omega_i \Leftrightarrow 2(\mathbf{m}_j-\mathbf{m}_i)^T\mathbf{x} \leq \mathbf{m}_j^T\mathbf{m}_j - \mathbf{m}_i^T\mathbf{m}_i \qquad \forall j \\
d^*(\mathbf{x}) &= \omega_i \Leftrightarrow 2(\mathbf{m}_j-\mathbf{m}_i)^T\mathbf{x} \leq \boldsymbol{\mu}_j^T\boldsymbol{\mu}_j - \boldsymbol{\mu}_i^T\boldsymbol{\mu}_i \qquad \forall j.
\end{aligned}$$

Dans le cas particulier de deux classes ($k = 2$) avec $n_1 = n_2 = 1$, on obtient

$$\begin{aligned}
d^*(\mathbf{x}) &= \omega_1 \Leftrightarrow 2(x_2 - x_1) \leq \boldsymbol{\mu}_2^T\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^T\boldsymbol{\mu}_1 = 0 \\
&\Leftrightarrow x_2 \leq x_1.
\end{aligned}$$

Les zones d'acceptation des classes ω_1 et ω_2 sont donc des demi-plans séparés par la droite d'équation $x_2 = x_1$.

2) Probabilité d'erreur (2pts + 1pt pour commentaire)

La probabilité d'erreur du classifieur est définie par

$$\begin{aligned}
P_e &= P(\omega_1) P[d^*(\mathbf{x}) = \omega_2 | \mathbf{x} \in \omega_1] + P(\omega_2) P[d^*(\mathbf{x}) = \omega_1 | \mathbf{x} \in \omega_2] \\
&= \frac{1}{2} P\left[2(\mathbf{m}_1-\mathbf{m}_2)^T\mathbf{x} \leq \boldsymbol{\mu}_1^T\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T\boldsymbol{\mu}_2 | \mathbf{x} \sim \mathcal{N}(\mathbf{m}_1, \mathbf{\Sigma}_1)\right] \\
&\quad + \frac{1}{2} P\left[2(\mathbf{m}_1-\mathbf{m}_2)^T\mathbf{x} \geq \boldsymbol{\mu}_1^T\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T\boldsymbol{\mu}_2 | \mathbf{x} \sim \mathcal{N}(\mathbf{m}_2, \mathbf{\Sigma}_2)\right].
\end{aligned}$$

En utilisant le rappel, si $\mathbf{x} \sim \mathcal{N}(\mathbf{m}_1, \mathbf{\Sigma}_1)$ alors

$$\begin{aligned}
(\mathbf{m}_1-\mathbf{m}_2)^T\mathbf{x} &\sim \mathcal{N}\left((\mathbf{m}_1-\mathbf{m}_2)^T\mathbf{m}_1, (\mathbf{m}_1-\mathbf{m}_2)^T\mathbf{\Sigma}_1(\mathbf{m}_1-\mathbf{m}_2)\right) \\
&= \mathcal{N}\left((\mathbf{m}_1-\mathbf{m}_2)^T\mathbf{m}_1, (\mathbf{m}_1-\mathbf{m}_2)^T\mathbf{\Sigma}_1(\mathbf{m}_1-\mathbf{m}_2)\right) \\
&= \mathcal{N}\left(\|\boldsymbol{\mu}_1\|^2, \sigma^2\left[\|\boldsymbol{\mu}_1\|^2 + \|\boldsymbol{\mu}_2\|^2\right]\right)
\end{aligned}$$

et si $\mathbf{x} \sim \mathcal{N}(\mathbf{m}_2, \mathbf{\Sigma}_2)$ alors

$$(\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{x} \sim \mathcal{N}\left(-\|\boldsymbol{\mu}_2\|^2, \sigma^2\left[\|\boldsymbol{\mu}_1\|^2 + \|\boldsymbol{\mu}_2\|^2\right]\right).$$

On en déduit

$$\begin{aligned}
P_e &= \frac{1}{2}P\left[(\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{x} \leq \frac{\|\boldsymbol{\mu}_1\|^2 - \|\boldsymbol{\mu}_2\|^2}{2} \,|\mathbf{x} \sim \mathcal{N}(\mathbf{m}_1, \mathbf{\Sigma}_1)\right] \\
&\quad + \frac{1}{2}P\left[(\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{x} \geq \frac{\|\boldsymbol{\mu}_1\|^2 - \|\boldsymbol{\mu}_2\|^2}{2} \,|\mathbf{x} \sim \mathcal{N}(\mathbf{m}_2, \mathbf{\Sigma}_2)\right] \\
&= \frac{1}{2}P\left[U = \frac{(\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{x} - \|\boldsymbol{\mu}_1\|^2}{\sigma\sqrt{\|\boldsymbol{\mu}_1\|^2 + \|\boldsymbol{\mu}_2\|^2}} \leq \frac{-\sqrt{\|\boldsymbol{\mu}_1\|^2 + \|\boldsymbol{\mu}_2\|^2}}{2\sigma} \,|U \sim \mathcal{N}(0,1)\right] \\
&\quad + \frac{1}{2}P\left[V = \frac{(\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{x} + \|\boldsymbol{\mu}_2\|^2}{\sigma\sqrt{\|\boldsymbol{\mu}_1\|^2 + \|\boldsymbol{\mu}_2\|^2}} \geq \frac{\sqrt{\|\boldsymbol{\mu}_1\|^2 + \|\boldsymbol{\mu}_2\|^2}}{2\sigma} \,|V \sim \mathcal{N}(0,1)\right].
\end{aligned}$$

En utilisant la symétrie de la loi normale, on obtient

$$P_e = F\left(\frac{-1}{2\sigma}\left[\sqrt{\|\boldsymbol{\mu}_1\|^2 + \|\boldsymbol{\mu}_2\|^2}\right]\right).$$

Comme F est une fonction croissante, lorsque σ diminue, la probabilité d'erreur diminue. En effet, lorsque la variance des lois conditionnelles à chaque classe est faible, il est facile de distinguer les différentes classes. Lorsque $\|\boldsymbol{\mu}_1\|$ et $\|\boldsymbol{\mu}_2\|$ augmentent, les moyennes des deux classes sont de plus en plus différentes et donc il est également plus facile de distinguer les deux classes. Ceci se traduit par une probabilité d'erreur qui diminue lorsque $\|\boldsymbol{\mu}_1\|$ et $\|\boldsymbol{\mu}_2\|$ augmentent.

3) **Estimation paramétrique (2pt)**

Danss les applications pratiques, on peut estimer $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ et $\boldsymbol{\sigma}^2$ à l'aide de leurs estimateurs du maximum de vraisemblance

$$\begin{aligned}
\widehat{\boldsymbol{\mu}}_1 &= \frac{1}{n_1}\sum_{i=1}^{n_1} \mathbf{t}^{(i)},\ \widehat{\boldsymbol{\mu}}_2 = \frac{1}{n_2}\sum_{i=1}^{n_2} \mathbf{u}^{(i)} \\
\widehat{\sigma^2} &= \frac{1}{n_1 + n_2}\left[\sum_{i=1}^{n_1}\left\|\mathbf{t}^{(i)} - \widehat{\boldsymbol{\mu}}_1\right\|^2 + \sum_{i=1}^{n_2}\left\|\mathbf{u}^{(i)} - \widehat{\boldsymbol{\mu}}_2\right\|^2\right]
\end{aligned}$$

4) **Questions de cours (1 pt par question = 6pts)**

- 4.1) Expliquer le principe de l'analyse en composantes principales. Comment peut-on choisir le nombre d'axes principaux ?

  Réponse : l'analyse en composantes principales est un prétraitement qui consiste à projeter les vecteurs de données sur les vecteurs propres associés aux valeurs propres les plus grandes de leur matrice de covariance. Le nombre d'axes principaux est le nombre de vecteurs propres sur lesquels on projette les données. Si $\lambda_1, ..., \lambda_p$ sont les valeurs propres de la matrice de covariance, on considère en général les vecteurs propres associés aux q valeurs propres les plus grandes, par exemple de manière à satisfaire

  $$\frac{\sum_{i=1}^{q}\lambda_i}{\sum_{i=1}^{p}\lambda_i} > 0.90.$$

- 4.2) Soit $\mathbf{T}$ la matrice de covariance d'un ensemble de données expertisées et $\mathbf{B}, \mathbf{S}$ les matrices de covariances inter-classe et intra-classe. Expliquer pourquoi les vecteurs propres de $\mathbf{T}^{-1}\mathbf{B}$ et de $\mathbf{S}^{-1}\mathbf{B}$ sont les mêmes. Dans les applications pratiques, est-il plus simple de rechercher les vecteurs propres et valeurs propres de $\mathbf{T}^{-1}\mathbf{B}$ ou de $\mathbf{S}^{-1}\mathbf{B}$? Pourquoi ?

  Réponse : puisque $\mathbf{T} = \mathbf{B} + \mathbf{S}$, si $\mathbf{u}$ est un vecteur propre de $\mathbf{S}^{-1}\mathbf{B}$, alors $\mathbf{Bu} = \lambda\mathbf{Su}$, d'où $\mathbf{Bu} = \lambda(\mathbf{T} - \mathbf{B})\mathbf{u}$ et donc

  $$(1+\lambda)\mathbf{Bu} = \lambda\mathbf{Tu} \text{ d'où } \mathbf{T}^{-1}\mathbf{Bu} = \frac{\lambda}{1+\lambda}\mathbf{u}$$

  donc $\mathbf{u}$ est aussi un vecteur propre de $\mathbf{T}^{-1}\mathbf{B}$. Inversement, si $\mathbf{u}$ est un vecteur propre de $\mathbf{T}^{-1}\mathbf{B}$, alors $\mathbf{u}$ est aussi un vecteur propre de $\mathbf{S}^{-1}\mathbf{B}$. Comme les valeurs propres de $\mathbf{T}^{-1}\mathbf{B}$ appartiennent à l'intervalle $[0, 1]$ (elles sont de la forme $\lambda/(1+\lambda)$), la recherche des valeurs propres de $\mathbf{T}^{-1}\mathbf{B}$ est en général numériquement plus facile que la recherche des valeurs propres de $\mathbf{S}^{-1}\mathbf{B}$.

- 4.3) On désire construire une fonction discriminante linéaire pour un problème de classification à deux classes. Quelle fonction de coût $J(\mathbf{w})$ utilise-t-on pour obtenir l'algorithme LMS ? Pourquoi ne peut-on pas utiliser la fonction de coût du filtre de Wiener-Hopf dans la plupart des applications pratiques ?

  Réponse : la fonction de coût utilisée pour obtenir l'algorithme LMS est

  $$J(\mathbf{w}) = \frac{1}{2}e^2(n)$$

  où $e(n) = d(n) - \mathbf{w}^T\mathbf{x}(n)$ est l'erreur instantanée. On ne peut pas utiliser la fonction de coût du filtre de Wiener-Hopf dans la plupart des applications pratiques car celle ci nécessite de connaître les autocorrélations des composantes du vecteur $\mathbf{x}(n)$ et les intercorrélations entre les composantes du vecteur $\mathbf{x}(n)$ et $d(n)$ qui sont rarement connues

- 4.4) Dans la théorie des machines à vecteurs supports, qu'est ce que la marge de la base d'apprentissage et qu'appelle-t-on vecteur support ?

  Réponse : la marge de la base d'apprentissage est la distance algébrique la plus faible entre tous les éléments de la base d'apprentissage et l'hyperplan séparateur. Un vecteur support est un vecteur dont la distance à l'hyperplan séparateur est égal à la marge.

- 4.5) Pourquoi est-il utile d'utiliser un prétraitement non-linéaire avant d'appliquer la théorie des machines à vecteurs supports ? Expliquer ce qu'est un noyau de Mercer et son intérêt pour les machines à vecteurs supports.

  Réponse : utiliser un prétraitement linéaire permet de résoudre des problèmes qui ne sont pas linéairement séparables mais qui sont non-linéairement séparables. Un noyau de Mercer est une fonction $k(x, y)$ qui peut s'écrire

  $$k(x, y) = \phi^T(x)\phi(y).$$

  L'utilisation d'un tel noyau permet de simplifier considérablement l'utilisation des machines à vecteurs support, puisqu'il suffit de remplacer tout produit scalaire $x^T y$ utilisé dans une analyse linéaire par $k(x, y)$ dans le cas non-linéaire sans être obligé de connaître la fonction ϕ.

- 4.6) Quel algorithme utilise-t-on pour mettre à jour les poids d'un réseau de neurones ? Qu'est ce que le théorème d'approximation universelle de Cybenko ?

  Réponse : l'algorithme utilisé pour mettre à jour les poids d'un réseau de neurones s'appelle l'algorithme de rétropropagation du gradient. Le théorème d'approximation universelle de Cybenko dit que toute fonction continue (non-linéaire) peut-être approchée avec la précision voulue par la sortie d'un réseau de neurones à une couche.

5) **Questions portant sur l'article (8pts)**

- 5.1) Dans l'application proposée dans l'article, comment est constitué le vecteur y ? (0.5pt)

  Réponse : Le vecteur y contient tous les pixels de l'image étudiée

- 5.2) Quelle est la motivation pour résoudre le problème (l^0) défini par l'équation (5) ? (0.5pt)

  Réponse : Le problème (l^0) cherche à minimiser le nombre d'éléments non nuls du vecteur $\mathbf{x}$ sous la contrainte $\mathbf{Ax} = \mathbf{y}$. La solution de ce problème sera une solution **parcimonieuse** de l'équation $\mathbf{Ax} = \mathbf{y}$, c'est-à-dire une solution $\mathbf{x}$ qui contient beaucoup de composantes nulles.

- 5.3) Pourquoi préfère-t-on résoudre le problème (l^1) défini par l'équation (6) au problème (l^0) défini par l'équation (5) ? (0.5pt)

  Réponse : Il existe des algorithmes simples comme les algorithmes de programmation linéaire permettant de résoudre le problème (l^1) alors qu'il n'existe pas d'algorithme simple permettant de résoudre le problème (l^0) directement.

- 5.4) Pourquoi doit-on en pratique remplacer le problème (l^1) défini par l'équation (6) par le problème (l_s^1) défini par l'équation (10) ? (0.5pt)

  Réponse : La présence de bruit dans l'image fait que la relation $\mathbf{Ax} = \mathbf{y}$ ne peut être vérifiée. Par contre, si le bruit est faible on pourra avoir $\|\mathbf{Ax} - \mathbf{y}\| \leq \varepsilon$.

- 5.5)Expliquer avec soin la règle de classification (12) (développer la réponse) (2pts)

  Réponse : Etant donné une image test, on calcule son approximation parcimonieuse en résolvant le problème (l_s^1) défini par l'équation (10). On obtient un vecteur $\widehat{\mathbf{x}}_1$ qui idéalement ne contient des éléments non nuls qu'aux positions associées à la classe de $\mathbf{y}$. Par exemple, si $\mathbf{y}$ est un vecteur de la classe ω_1, les seuls éléments non-nuls de $\widehat{\mathbf{x}}_1$ devraient faire partie des n_1 premières composantes de $\widehat{\mathbf{x}}_1$. Si $\mathbf{y}$ est un vecteur de la classe ω_2, les seuls élements non-nuls de $\widehat{\mathbf{x}}_1$ sont certains éléments $\widehat{\mathbf{x}}_1(k)$ avec $k \in \{n_1 + 1, ..., n_1 + n_2\}$ etc ...La fonction caractéristique de $\widehat{\mathbf{x}}_1$ notée $\delta_i(\widehat{\mathbf{x}}_1)$ garde les composantes $\widehat{\mathbf{x}}_1(k)$ avec $k = n_{i-1} + 1, ..., n_{i-1} + n_i$ et met à zéro toutes les autres composantes de $\widehat{\mathbf{x}}_1$. On calcule ensuite l'erreur de reconstruction entre le vecteur $\mathbf{y}$ et le vecteur $\mathbf{A}\delta_i(\widehat{\mathbf{x}}_1)$. On associe le vecteur $\mathbf{y}$ à la classe associée à la plus petite erreur de reconstruction, c'est-à-dire

$$d(\mathbf{y}) = \omega_i \Longleftrightarrow \|\mathbf{y} - \mathbf{A}\delta_i(\widehat{\mathbf{x}}_1)\| \leq \|\mathbf{y} - \mathbf{A}\delta_j(\widehat{\mathbf{x}}_1)\|, \qquad \forall j = 1, ..., k$$

- 5.6) Comment voit-on sur la figure 3 que la méthode proposée marche bien pour l'image test ? Inversement, comment voit-on sur la figure 4 que la méthode basée sur une minimisation l_2 ne marche pas pour l'image test ?(1pt)

Réponse : Lorsque l'image test est associée à un des individus de la base d'apprentissage, la solution $\widehat{\mathbf{x}}_1$ est parcimonieuse et l'erreur de reconstruction (appelée résidu) est faible pour la classe de $\mathbf{y}$. Sur la figure 3 (a), on voit que seuls quelques coefficients de $\widehat{\mathbf{x}}_1$ sont de grande valeur et que l'erreur de reconstruction associée à la classe ω_1 (qui contient les valeurs importantes de $\widehat{\mathbf{x}}_1$) est beaucoup plus faible que les erreurs de reconstructions associées aux autres classes $\omega_2, ..., \omega_k$. Inversement sur la figure 4, la solution $\widehat{\mathbf{x}}_1$ n'est pas parcimonieuse et il n'y a pas d'erreur de reconstruction beaucoup plus petite que les autres. D'ailleurs la méthode déciderait d'affecter $\mathbf{y}$ à la classe ω_{15} alors que la classe correcte est ω_1.

- 5.7) A quoi sert la règle de décision (15) ? (0.5pt)

  Réponse : La règle de décision (15) permet de décider si l'image test correspond à une des classes de la base d'apprentissage ($\text{SCI}(\widehat{\mathbf{x}}) \geq \tau$) ou si l'image test est une image aberrante ($\text{SCI}(\widehat{\mathbf{x}}) < \tau$).

- 5.8) En présence d'occlusion, que représente le vecteur $\mathbf{e}_0$ dans (19) ? (0.5pt)

  Réponse : Le vecteur $\mathbf{e}_0$ dans (19) contient les erreurs dues aux pixels situés dans la zone d'occlusion. Plus précisément, $e_0(i) = 0$ si le $i^{ème}$ pixel de l'image n'est pas dans une zone d'occlusion et $e_0(i)$ est l'erreur entre le pixel que l'on observe et celui qu'on devrait observer si le $i^{ème}$ pixel de l'image est dans une zone d'occlusion.

- 5.9) A quelle condition sur $\mathbf{e}_0$ peut on estimer $\mathbf{w}_0 = \left[\mathbf{x}_0^T, \mathbf{e}_0^T\right]^T$ sans erreur ? (0.5pt)

  Réponse : Il faut que $\mathbf{e}_0$ soit un vecteur suffisamment parcimonieux.

- 5.10) Expliquer la forme du résidu $r_i(y)$ dans (23) (1pt)

  Réponse : y_r est le pixel de l'image pour lequel on a corrigé l'occlusion. On utilise alors la même rêgle que (12) mais avec une éventuelle correction qui dépend de la présence ou de l'absence de l'occlusion.

- 5.11) Comment les auteurs ont-ils simulés une occlusion ? (0.5pt)

  Réponse : ils ont remplacé la valeur d'un pixel sujet à une occlusion par une valeur uniformément répartie dans l'ensemble $\{0, ..., 255\}$.