le cnam

Sujet UE RCP208
Reconnaissance des formes et méthodes neuronales

Année universitaire 2016–2017

Examen 1ère session : 1er juillet 2017

Responsable : Michel CRUCIANU

Durée : 2h00

Seul document autorisé : 2 pages A4 recto-verso, manuscrites.

Les téléphones mobiles et autres équipements communicants (PC, tablette, etc.) doivent être éteints et rangés dans les sacs pendant toute la durée de l'épreuve.

Sujet de 8 pages, celle-ci comprise.

Vérifiez que vous disposez bien de la totalité des pages du sujet en début d'épreuve et signalez tout problème de reprographie le cas échéant.

---

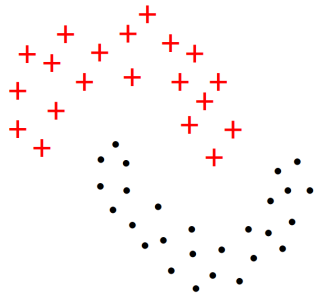
1. Expliquez brièvement, avec vos propres mots, pourquoi l'erreur d'apprentissage d'un modèle n'est pas nécessairement représentative de son erreur de généralisation. (**2 points**)

Correction **:**

Un modèle trop complexe peut « apprendre par cœur » les données d'apprentissage, c'est à dire s'adapter aux particularités de l'échantillon d'apprentissage, en s'éloignant ainsi des caractéristiques plus générales du problème, avec pour conséquence des performances de généralisation diminuées (voir le cours d'introduction).

---

2. Considérons les données bidimensionnelles ($\in \mathbb{R}^2$) de la figure suivante, chacune appartenant à une de deux classes (la classe des $+$ et la classe des $\cdot$).



— Indiquez (approximativement, en reproduisant la forme des nuages sur votre copie) la direction du premier axe principal d'une ACP normée et la direction du premier axe discriminant. (**2 points**)

— Combien y a-t-il d'axes discriminants ? Et si nous avions analysé des données appartenant à deux classes mais situées dans un espace de dimension 10 ($\in \mathbb{R}^{10}$) ? Justifiez brièvement. (**2 points**)

— Indiquez un résultat probable de l'estimation d'un mélange de 4 lois normales bidimensionnelles (sans contraintes sur les matrices de variances-covariances) avec ces données, en dessinant des ellipses illustrant les 4 lois normales (une seule ellipse par loi). (**2 points**)

— Peut-on déduire (approximativement) de ce mélange quel serait le résultat de l'application de l'algorithme *K-means* avec $k = 4$? A quelle condition ? (**2 points**)

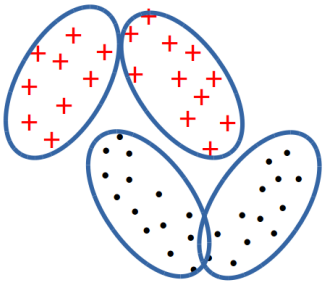
Correction **:**

— Le premier axe principal et l'axe discriminant ont approximativement la même orientation, bas-droite - haut-gauche. Pour trouver l'axe discriminant il faut appliquer la

déformation correspondant à la matrice des covariances empiriques totales et ensuite regarder les centres de gravité des deux classes (mais ce niveau de précision n'était pas exigé).

— Pour un problème de discrimination entre deux classes il ne peut y avoir qu'un seul axe discriminant, car la matrice E (covariances inter-classes) n'est calculée qu'à partir de 2 observations (les centres de gravité des 2 classes), donc son rang est 1 (il y a une seule valeur propre non nulle).

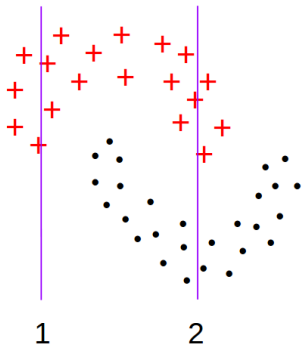
— Un résultat très probable est indiqué dans la figure suivante. Il correspond à un maximum global de la vraisemblance pour $m = 4$ composantes (sachant toutefois que l'algorithme EM peut converger vers un maximum local).



— Les groupes obtenus par *K-means* avec $k = 4$ devraient correspondre approximativement aux sous-ensembles de données mieux expliquées par une des lois que par les autres. A condition que les lois ne soient pas trop « déformées » (ellipses trop allongées) car *K-means* avec la métrique euclidienne classique a tendance à trouver des groupes plutôt ronds (pas trop allongés).

---

3. Supposons que parmi les observations (données) du problème précédent il y a deux observations incomplètes, pour lesquelles seules les abscisses sont connues. Elles sont indiquées dans la figure suivante par les traits verticaux 1 et 2 (l'ordonnée étant inconnue). Entre l'imputation par la moyenne et l'imputation par les *k* plus proches voisins, laquelle préférez-vous dans ce cas et pourquoi ? (**2 points**)



Correction **:**

On préfère l'imputation par les *k* plus proches voisins car l'imputation par la moyenne tient compte uniquement de la distribution marginale des ordonnées, alors qu'ici on observe une dépendance forte entre l'abscisse et l'ordonnée.

---

4. On s'intéresse au paramètre T (la température) utilisé dans l'apprentissage des cartes topologiques. Supposons qu'on utilise une carte 2D de 7×7 (msize= [7 7] dans somtoolbox), une fonction de voisinage de type gaussien (Neigh = 'gaussian' dans somtoolbox) et que les poids de la carte son initialisés aléatoirement (avec "rand_init" dans somtoolbox).

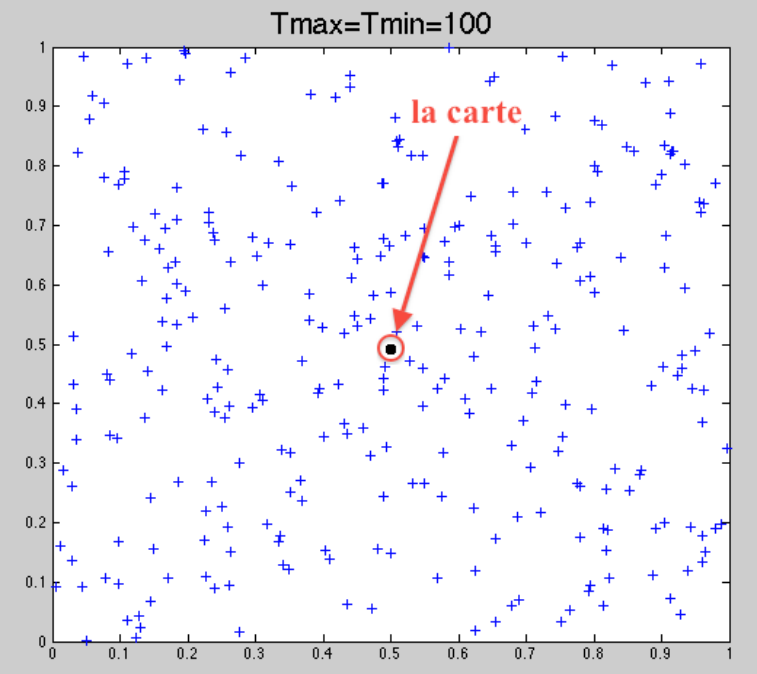
   (a) On décide de faire un apprentissage avec une **température constante**, c'est-à-dire un apprentissage avec une seule phase et Tmax=Tmin=T (radius_ini=radius_fin dans somtoolbox). Décrivez la différence entre les cartes obtenues à la fin de l'apprentissage pour les cas suivants : Tmax=Tmin=100, Tmax=Tmin=3 et Tmax=Tmin=0.01. (**2 points**)

   (b) On décide de faire décroître la valeur de T dans un intervalle (comme en TP). La convergence vers la solution peut alors se décomposer en deux phases : la première correspond aux grandes valeurs de T et la deuxième a lieu pour les petites valeurs de T. Décrivez brièvement le comportement de la carte durant chacune de ces deux phases. (**2 points**)

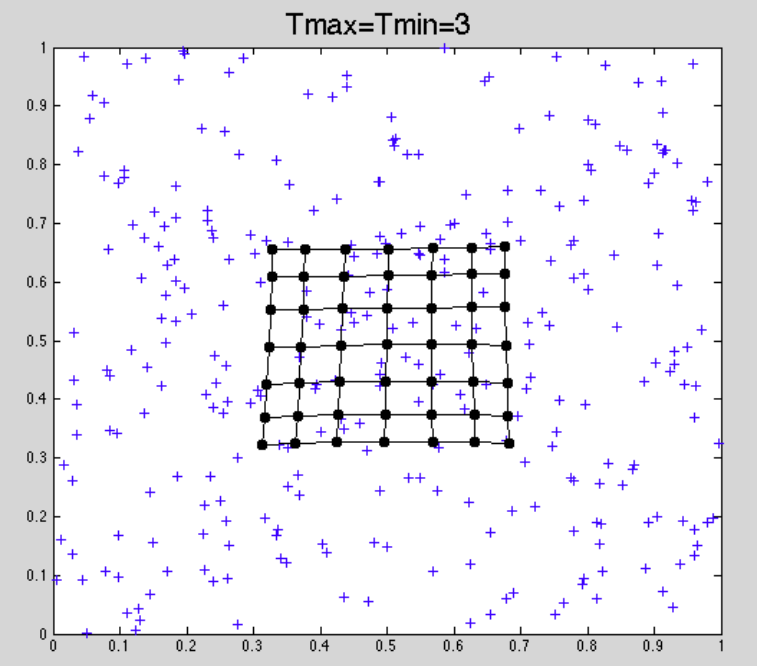
Correction **:**

(a) On décide de faire un apprentissage avec une **température constante**, c'est-à-dire un apprentissage avec une seule phase et Tmax=Tmin=T (radius_ini=radius_fin dans somtoolbox). Décrivez la différence entre les cartes obtenues à la fin de l'apprentissage pour les cas suivants : Tmax=Tmin=100, Tmax=Tmin=3 et Tmax=Tmin=0.01.

— Tmax=Tmin=100
La température est trop élevée par rapport à la taille de la carte, tous les neurones sont attirés vers le centre de gravité du nuage de points (figure ci-dessous)

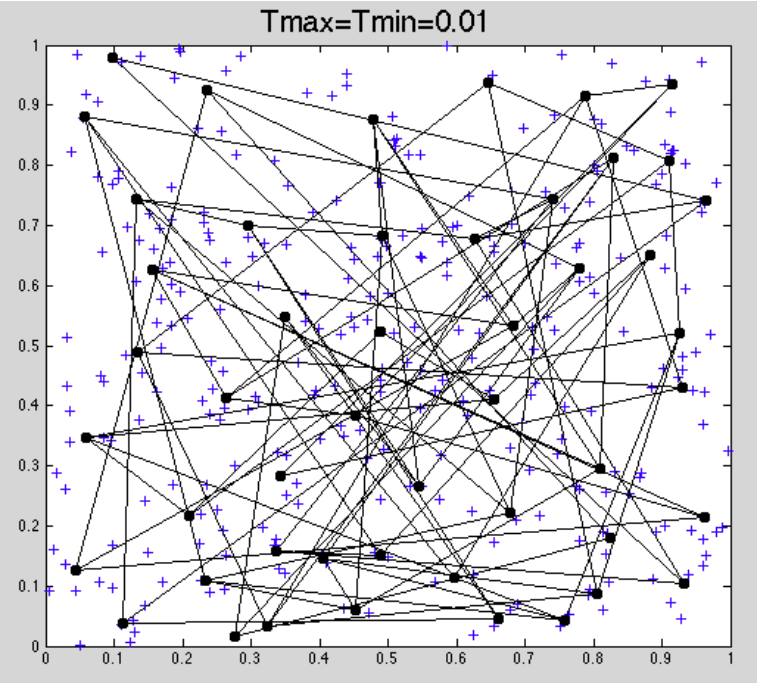


— Tmax=Tmin=3
La température est (raisonnablement) élevée par rapport à la taille de la carte. L'ordre topologique se forme mais la carte ne parvient pas à se déployer sur l'ensemble des données (figure ci-dessous)



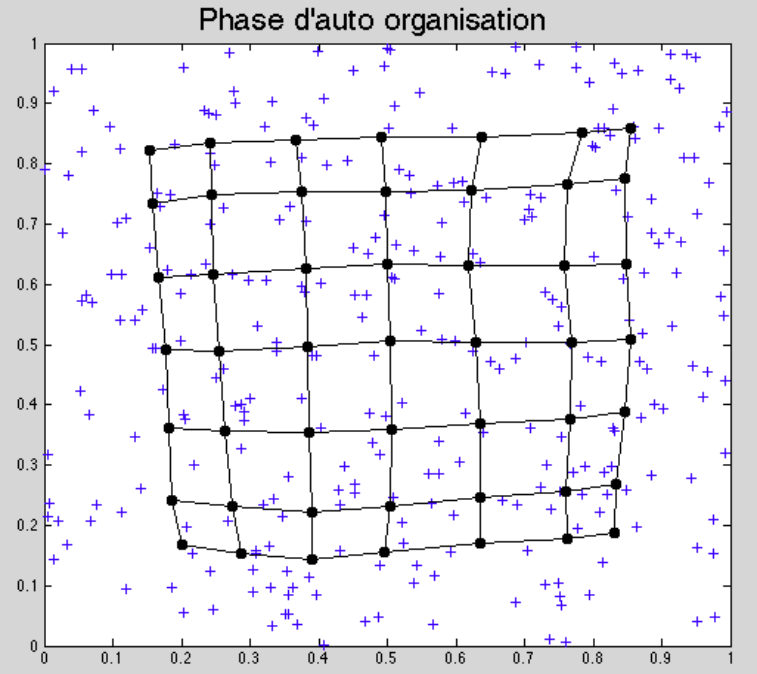
— Tmax=Tmin=0.01
La température est trop petite (c'est l'algorithme des k-moyennes). Les référents se répartissent plus finement sur les données mais la carte obtenue n'est pas ordonnée

(l'ordre topologique n'est pas respecté, voir la figure ci-dessous)

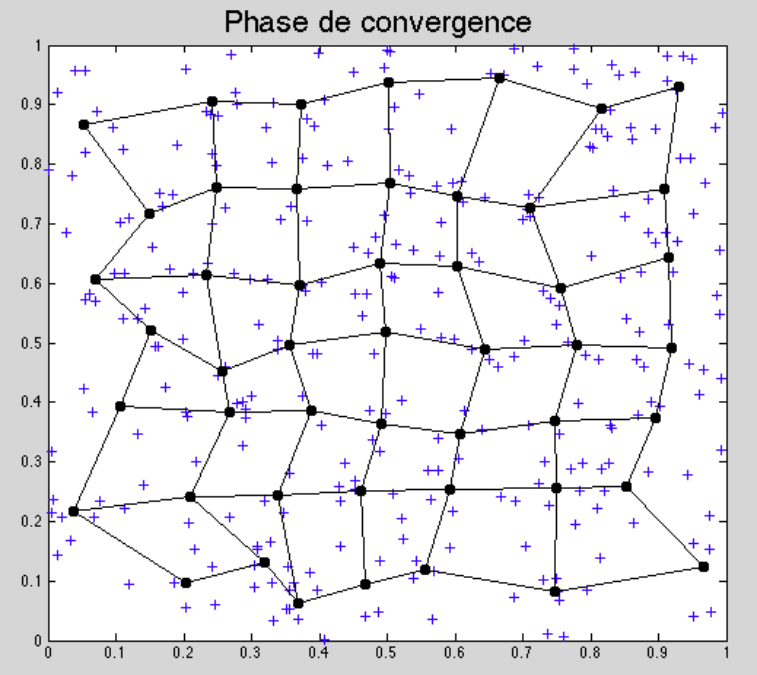


(b) On décide de faire décroître la valeur de T dans un intervalle (comme en TP). La convergence vers la solution peut alors se décomposer en deux phases : la première correspond aux grandes valeurs de T et la deuxième a lieu pour les petites valeurs de T. Décrivez brièvement le comportement de la carte durant chacune de ces deux phases.

— La première phase correspond aux grandes valeurs de T. Elle a tendance à assurer la conservation de l'ordre topologique (l'ordre topologique se forme). Plus la valeur de T diminue, plus la carte se déploie.



— La deuxième phase a lieu pour les petites valeurs de T. L'algorithme commence à se rapprocher de l'algorithme des k-moyennes. A la fin de l'algorithme, quand la valeur de T devient plus petite, les référents se répartissent plus finement sur les données.



5. On veut résoudre un problème de classification non linéairement séparable à l'aide d'un per-

ceptron multicouches. Les formes à classer sont en dimension $\mathbf{m}$ et il y a $\mathbf{p}$ classes. Le réseau de neurones utilisé possède une couche cachée de $\mathbf{n}$ neurones. La fonction d'activation des neurones cachés est une tangente hyperbolique.

(a) Expliquer le codage utilisé pour les sorties. (**1 point**)

(b) Quel est le nombre de poids (biais + poids) de ce réseau ? (**1 point**)

(c) Quelle est la fonction d'activation utilisée pour les neurones de la couche de sortie ? Comment peut-on interpréter les sorties du réseau ? (**1 point**)

(d) Pourquoi il est nécessaire de normaliser les entrées (que se passe-t-il si on a une entrée très grande) ? (**1 point**)

Correction :

(a) Expliquer le codage utilisé pour les sorties.

Le codage classique des sorties désirées pour la classification utilise un neurone de sortie par classe, avec une valeur désirée haute pour le neurone de la classe correcte, et une valeur désirée faible pour les autres classes. On peut par exemple utiliser le codage 1 parmi p, c'est-à-dire que lorsqu'une entrée appartient à la classe i, le neurone de sortie s_i correspondant devrait se positionner à 1, et les autres à -1 (ou 0).
Par exemple, pour les Iris de Fisher (donc p=3), on peut utiliser le codage suivant :

classe 1 $\longrightarrow$ **(1 0 0)** **classe 2** $\longrightarrow$ **(0 1 0)** **classe 3** $\longrightarrow$ **(0 0 1)**

(b) Quel est le nombre de poids (biais + poids) de ce réseau ?

$mn + np + n + p$

(c) Quelle est la fonction d'activation utilisée pour les neurones de la couche de sortie ? Comment peut-on interpréter les sorties du réseau ?

Pour un problème de classification il vaut mieux utiliser des fonctions d'activation non linéaires ('logistic' ou 'softmax') à la couche de sortie. Quand on utilise la fonction d'activation 'softmax' la somme des sorties est égale à 1. On peut ainsi interpréter les sorties du réseau comme des probabilités que l'entrée appartienne à chacune des classes.

(d) Pourquoi il est nécessaire de normaliser les entrées (que se passe-t-il si on a une entrée très grande) ?

La fonction d'activation des neurones cachés est une tangente hyperbolique. Il est impératif que les entrées soient de moyenne pas trop loin de zéro et de variance pas trop loin de 1. Si on a une entrée très grande, elle fait saturer plusieurs neurones et bloque l'apprentissage pour cet exemple.
