Contrôle de Connaissance
Master Recherche Information, Apprentissage, Cognition - Université Paris-Sud 11
Module « Apprentissage Statistique et Optimisation »

Lundi 21 novembre 2011
Durée : 3h00
Documents autorisés : supports et notes de cours

Les exercices sont indépendants.
Rédigez les parties I et II sur des copies différentes pour faciliter la correction.

Partie I

1 Classification de documents (10 pts)

Dans un cadre de classification de documents, on décide de représenter chaque document d par un « sac-de-mot », c'est-à-dire par un vecteur n_W-dimensionnel x_d dont chaque composante x_{dw} correspond au nombre d'occurrences du mot w dans le document d.

1. On choisit de modéliser ce vecteur de comptes indépendamment mot par mot. Le nombre d'occurrences du mot w dans d est une variable aléatoire X_{dw} qui suit une loi paramétrée par $\beta_w \in [0, 1]$ exprimant la probabilité que w apparaisse k fois dans un document d de longueur l_d selon :

$$\begin{cases} P(X_{dw} = k; \beta_w, l_d) = \binom{l_d}{k} \beta_w^k (1 - \beta_w)^{l_d - k} \text{ si } k \leq l_d \\ P(X_{dw} = k; \beta_w, l_d) = 0 \text{ si } k > l_d \end{cases}$$

   Ce modèle est appelé dans la suite le modèle *binomial*.

   (a) Sous l'hypothèse que les composantes du vecteur X_d sont générées indépendamment les unes des autres, exprimez la probabilité complète d'un document $P(X_d; \beta, l_d)$, avec β le vecteur de dimension n_W dont les composantes sont β_w.

   (b) On veut estimer ce modèle à partir d'une collection de documents $d_1 \dots d_{n_D}$. Détaillez la démarche qui permet d'obtenir des estimateurs $\widehat{\beta_w}$ et donnez l'expression de ces estimateurs.

2. Le modèle *binomial* s'avérant trop simple pour décrire finement les répétitions des mots dans les textes, on décide de construire un modèle plus riche, correspondant à l'histoire générative suivante. Pour générer le document d, pour chaque mot w, on commence par décider si le mot apparait dans le vocabulaire sur lequel est construit d en tirant une variable de Bernoulli Z_w suivant une loi paramétrée par α_w. Si Z_w vaut 0 (ce qui arrive avec probabilité $(1 - \alpha_w)$), le mot n'apparait pas dans le document ($x_{dw} = 0$) ; si Z_w vaut 1 (avec probabilité α_w), *on fait un nouveau tirage*, en utilisant comme précédemment, une loi binomiale paramétrée par β_w. S'en déduit une valeur (qui peut être positive ou nulle) pour x_{dw}.

   (a) Écrire la loi que suit la variable X_{dw}, qui est maintenant régie par les deux paramètres α_w et β_w. On distinguera le cas $x_{dw} = 0$ du cas $x_{dw} > 0$.

   (b) On s'interesse maintenant à l'estimation des paramètres à partir d'une collection de documents $d_1 \dots d_{n_D}$. Écrire le programme d'optimisation auquel cette estimation donne lieu.

   Ce programme n'a pas de solution analytique. Dans la mesure où le modèle implique une variable non-observée (latente) Z_w, on imagine alors de mettre en place une stratégie itérative similaire à celle mise en œuvre dans l'algorithme EM.

   (c) L'étape E demande de calculer les probabilités *a posteriori* de Z_w sachant x_{dw}. Quand x_{dw} est strictement positif, il n'y a qu'une seule possibilité : $Z_w = 1$. On a donc de manière triviale $P(Z_w = 1 | x_{dw}; \alpha_w, \beta_w, l_d) = 1 = 1 - P(Z_w = 0 | x_{dw}; \alpha_w, \beta_w, l_d)$. Quand x_{dw} est nul, les choses sont moins évidentes. La valeur observée

$x_{dw} = 0$ peut être dûe soit au premier tirage, pour lequel on aurait obtenu $Z_w = 0$, soit au second tirage dans la loi binomiale si on a au contraire eu $Z_w = 1$. Donnez l'expression de $p_1(w, d) = P(Z_w = 1|X_{dw} = 0; \alpha_w, \beta_w, l_d)$ et de $p_0(w, d) = P(Z_w = 0|X_{dw} = 0; \alpha_w, \beta_w, l_d)$.

(d) L'étape M demande de maximiser en α_w et β_w la fonction auxilaire Q définie comme l'espérance de la log-vraisemblance complétée. De nouveau on peut traiter composante par composante, et pour chaque mot considérer la fonction suivante [1] :

$$Q_w(\alpha_w; \beta_w, l_d) = \sum_{d, x_{dw}>0} \log P(Z_w = 1, X_{dw} = x_{dw}) + \sum_{d, x_{dw}=0} p_0(w, d) \log P(Z_w = 0, X_{dw} = 0) + p_1(w, d) \log P(Z_w = 1, X_{dw} = 0)$$

En injectant dans cette équation les valeurs de $P(Z_w, X_{dw} = x_{dw}; \alpha_w, \beta_w, l_d)$, puis en écrivant les conditions d'optimalité, donnez les formules de réestimation pour les paramètres α_w et β_w.
Si vous avez l'intuition de ces formules de réestimation par analogie avec vos réponses à la première question, vous pouvez vous contenter de les donner directement sans justification.

2 Segmentation de séquences (10 pts)

On s'intéresse dans cet exercice à des problèmes de *segmentation* de séquences (ici, des séquences de lettres). On observe donc des mots sur un alphabet fini Σ contenant m symboles et on cherche à apprendre à les découper en fragments plus petits. Ces problèmes apparaissent typiquement lorsque l'on souhaite segmenter des phrases en mots (en l'absence de séparateur visible, comme l'espace), ou des mots en unités plus petites comme des morphèmes, ou encore pour traiter des séquences biologiques ou des séquences d'actions d'un robot. Pour les besoins de l'exercice, on considère que l'alphabet ne contient que deux lettres a et b (c'est-à-dire que l'on a $m = 2$).

La segmentation est modélisé par un HMM, dans lequel chaque état correspond à la longueur du segment émis. En d'autres termes, dans l'état i, on ne peut émettre que des segments de longueur i : des lettres isolées dans l'état 1, des couples de lettres dans l'état 2, etc. La loi d'émission associée à i ne donne une probabilité non nulle qu'à des segments de Σ^i. Soit alors un mot de 4 lettres comme $abba$: dire que ce mot est émis le long de la séquence d'états 112 équivaut donc à postuler la segmentation $a|b|ba$.

1. On considère qu'un fragment ne peut pas contenir plus de n symboles et on s'intéresse donc à des HMM possédant n états.
   (a) Dessinez le HMM correspondant au cas $n = 2$, en indiquant clairement le nombre de paramètres qu'il contient ;
   (b) Plus généralement, donnez le nombre de paramètres du modèle en fonction de n et de m.
2. On considère maintenant le cas $n = 3$ et on fait l'hypothèse supplémentaire que la différence de longueurs de deux fragments successifs est au plus 1. Ainsi, par exemple, $abbbaba$ peut être segmenté $ab|bba|ba$, mais pas $a|bbb|ab|a$ car deux segments successifs a et bbb ont des différences de longueur trop grandes.
   (a) Dessinez le HMM correspondant et donnez son nombre de paramètres ;
   (b) Donnez le nombre de paramètres du modèle en fonction de n et de m ;
3. On revient au cas $n = 2$.
   (a) Un informateur vous communique l'échantillon d'apprentissage suivant composé de mots et leur segmentation correcte : $\{a|bb|ab|a, bb|a|ba|a|bb, b|a|bb|bb|a\}$. Expliquez quel principe vous utilisez pour estimer les paramètres et donnez la valeur des paramètres ainsi estimés.
   (b) En utilisant les estimateurs calculés à la question précédente, donnez la probabilité du mot $aaba$
   (c) Toujours avec les mêmes estimateurs, énumérez les segmentations de $bbba$ et calculez la segmentation la plus probable.
   (d) Votre informateur est prêt à vous fournir des nouveaux exemples de mots (mais sans la bonne segmentation). Expliquez comment vous pourriez les utiliser.
   (e) On souhaite maintenant faire dépendre les lois d'émission de la longueur du segment courant et de la longueur du segment précédent, ce qui revient à passer à une modélisation à l'ordre 2. Représentez le modèle correspondant.

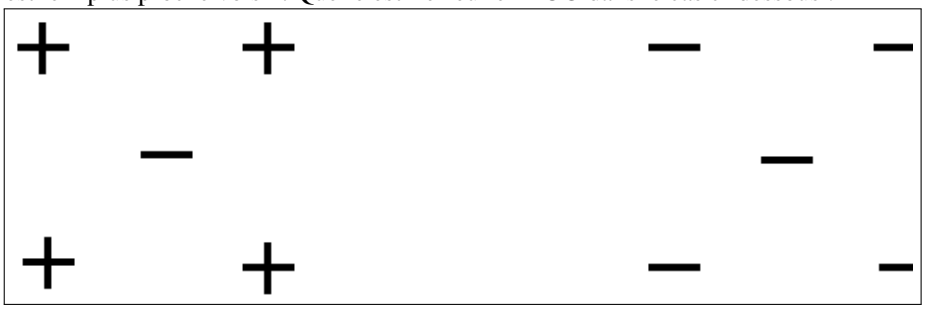
1. Pour alléger les notations, on omet ici les dépendances en α_w et β_w, mais elles existent !

Partie II

3 Validation (5 points)

- Qu'est-ce que le sur-apprentissage (overfitting) ? Comment le détecte-t-on ?
- Vous apprenez un réseau de neurones à p neurones sur la couche cachée. Doit-on augmenter ou diminuer p en cas d'overfitting ? Pourquoi ?
- La validation croisée découpe les données $\mathcal{E}$ en k sous-ensembles $\mathcal{E}_1, \ldots \mathcal{E}_k$ (k est souvent égal à 10). Une hypothèse h_i est apprise à partir de
$$\mathcal{E}_{-i} = \mathcal{E}_1 \bigcup \ldots \bigcup \mathcal{E}_{i-1} \bigcup \mathcal{E}_{i+1} \bigcup \ldots \bigcup \mathcal{E}_k$$
et on calcule son erreur sur $\mathcal{E}_i$. L'erreur de l'apprentissage est estimée par la moyenne des erreurs des h_i. Quelles précautions faut-il prendre lorsqu'on découpe les données ?
- Le leave-one-out (LOO) généralise la validation croisée : si $\mathcal{E}$ contient n exemples, on définit n sous-ensembles. L'estimation de l'erreur par LOO par rapport à la validation croisée est-elle
  - plus optimiste,
  - plus pessimiste
  - ni l'un ni l'autre ?

  Pourquoi ?
- Votre algorithme est le 1 plus proche voisin. Quelle est l'erreur en LOO dans le cas ci-dessous ?



Est-ce mieux de prendre un 3-plus proche voisin ?

4 Arbre de décision (5 points)

Vous avez observé des objets de deux classes ($+$ et $-$).

forme	taille	couleur	classe
rond	petit	blanc	$+$
carré	petit	rouge	$-$
rond	petit	vert	$+$
carré	moyen	blanc	$+$
rond	petit	rouge	$-$
rond	petit	jaune	$+$
rond	moyen	jaune	$-$
rond	moyen	blanc	$+$
carré	grand	blanc	$+$
carré	grand	rouge	$-$
carré	grand	vert	$-$

- Quelle quantité d'information est donnée par *couleur* = *blanc* ?
- Quel est le gain d'information de chaque attribut ?
- Déroulez l'algorithme d'arbre de décision et montrez l'arbre obtenu.
- Transformer l'arbre obtenu en une base de règles.
  Inversement, peut-on transformer toute base de règles en un arbre de décision ?
- A-t-on intérêt à faire de la sélection d'attributs avant d'apprendre un arbre de décision ?

5 Réduction de problèmes d'apprentissage (5 points)

Vous disposez d'un algorithme $\mathcal{A}$ d'apprentissage binaire (applicable quand les labels sont $\{0, 1\}$). Votre problème d'apprentissage comprend k classes. Construisez un algorithme $\mathcal{B}$ pour apprendre une hypothèse à k classes en utilisant $\mathcal{A}$ (plusieurs réponses possibles).
Supposons que l'erreur moyenne commise par $\mathcal{A}$ soit de 10%. Peut-on borner l'erreur de $\mathcal{B}$?

Vous disposez d'un algorithme d'estimation de densité $\mathcal{D}$, utilisant une base d'apprentissage sans label $\mathcal{E} = \{x_1, \dots x_n, x_i \in \mathbb{R}^d\}$. La base d'apprentissage est issue d'un échantillonnage, et l'estimation apprise donne pour tout ensemble $A \subset \mathbb{R}^d$ la probabilité qu'un échantillon appartienne à A.

Construisez un algorithme $\mathcal{C}$ pour apprendre une hypothèse à 2 classes en utilisant $\mathcal{D}$, en supposant que les deux classes sont d'égale probabilité. Comment tenir compte de la probabilité des deux classes ?

6 Optimisation (5 pts)

Nous considérons la minimisation de fonctions $f : \mathbf{x} \in \mathbb{R}^n \mapsto f(\mathbf{x}) \in \mathbb{R}$. Les coordonnées d'un vecteur $\mathbf{x}$ de $\mathbb{R}^n$ sont notées $\mathbf{x} = (x_1, \dots, x_n)$, i.e. pour tout i, $x_i \in \mathbb{R}$. Nous définissions les fonctions suivantes :

- $\star$ $f_{\text{elli}}(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n}(10^5)^{\frac{i-1}{n-1}}x_i^2$
- $\star$ $f_{g\circ f_{\text{elli}}}(\mathbf{x}) = g(f_{\text{elli}}(\mathbf{x}))$ avec $g : x \in \mathbb{R} \mapsto x^{1/4}$
- $\star$ $f_{\text{ellinoisy}}(\mathbf{x}) = \frac{1}{2}\left(\sum_{i=1}^{n}(10^5)^{\frac{i-1}{n-1}}x_i^2\right)\exp(10^{-5}\mathcal{N}(0,1))$ avec $\mathcal{N}(0,1)$ une variable aléatoire suivant une loi normale, tirée aléatoirement à chaque appel de fonction

1. Expliquer le principe général des algorithmes d'optimisation Broyden-Fletcher-Goldfarb-Shanno (BFGS) et Covariance Matrix Adaptation Evolution Strategy (CMA-ES).
2. Ces deux algorithmes sont testés sur le fonction f_{elli} en dimension $n = 5$. Le nombre d'évaluations pour attendre une valeur de fonction objectif inférieure à 10^{-6} pour 3 essais indépendants où le point initial est tiré aléatoirement est donné dans le tableau suivant

	Tests on f_{elli}					
	BFGS			CMA-ES		
# Evals to obtain a function value $\leq 10^{-6}$	484	510	599	2162	2218	2290

   Commenter et expliquer les différences observées.
3. Auriez vous observé une convergence plus lente ou plus rapide si vous aviez utilisé un algorithme de descente de gradient à la place de l'algorithme BFGS ? Même question si vous aviez pris un algorithme (1+1)-ES avec règles des $1/5$ à la place de CMA-ES. Justifier les réponses.
4. Nous testons maintenant dans les mêmes conditions expérimentales les algorithmes BFGS et CMA-ES sur la fonction $f_{g\circ f_{\text{elli}}}$. Nous obtenons les résultats suivants :

	Tests on $g \circ f_{\text{elli}}$					
	BFGS			CMA-ES		
# Evals to obtain a function value $\leq 10^{-6}$	3929	3979	4049	2658	2882	2794

   Expliquer les différences observées par rapport aux résultats obtenus sur f_{elli}.
5. En utilisant les mêmes conditions expérimentales et même critères d'arrêt que précédemment pour la fonction $f_{\text{ellinoisy}}$, nous obtenons les résultats suivants

	Tests on $f_{\text{ellinoisy}}$					
	BFGS			CMA-ES		
$\|\mathbf{x}_{\text{final}}\|$	0.9	0.56	3.69	2.13×10^{-8}	6.62×10^{-9}	2.81×10^{-8}

   où $\mathbf{x}_{\text{final}}$ est la solution retournée par l'algorithme.
   Qu'observez-vous ? Comment expliquer les différences observées entre les solutions retournées par BFGS et CMA-ES ?