TD 8

Exercice 1 – Clustering Spectral

On considère un graphe non orienté $G = (V, E)$ de n sommets, $V = \{v_1, \ldots, v_n\}$. On lui associe $W = (w_{ij})_{i,j=1,\ldots,n'}, w_{ij} \geq 0$ une matrice de poids non négative (symétrique). Les nœuds de ce graphe représentent les exemples à clusteriser, les poids représentent les similarités entre exemples : plus un poids entre deux nœuds est fort, plus les deux exemples associés sont similaires. Un des avantages du clustering spectral est donc de fonctionner sur une notion de similarité (comme les noyaux dans les SVMs) et non pas uniquement géométriquement sur une description euclidienne de nos exemples. Bien sûr, toute description euclidienne peut être représentée par une similarité (prendre l'exponentiel de l'inverse de la distance par exemple). Le principe du clustering spectral est de former des partitions de telle manière que les nœuds dans une même partition aient une connectivité très forte (les poids au sein de chaque cluster doivent être très élevés) et que les poids inter-clusters soient le plus faible possible - qu'il y ait une faible connectivité entre les sommets de deux clusters disjoints. En langage graphe, il s'agit de faire un partitionnement du graphe tel que la *coupe* associée fasse disparaître les connections de poids le plus faible possible.
La mesure de qualité d'une coupe en k partitions $A_1, \ldots, A_k$ est définie par :

$$Rcut(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{|A_i|}$$

avec $W(A, B) = \sum_{i \in A, j \in B} w_{ij}$, $\bar{A}$ le complément de A dans V et $|A|$ le nombre de sommets de A. Le clustering spectral cherche donc à résoudre : $min_{A_1,\ldots,A_k}\ Rcut(A_1, \ldots, A_k)$ avec $A_1, \ldots, A_k$ une partition des sommets de V.
L'algorithme se base sur la notion de Laplacien d'un graphe : soit W la matrice des poids et $D = (d_i)_{i=1\ldots n}$ la matrice diagonale telle que $d_i = \sum_{j=1}^{n} w_{ij}$, le Laplacien du graphe est la matrice $L = D - W$. Le but de l'exercice est d'examiner une interprétation en terme de partitionnement de graphe.

Rappel d'algèbre linéaire : Une matrice $M \in \mathbb{R}^{d \times d}$ symétrique est dite semi-définie positive si elle vérifie entre autre l'une de ces propriétés (équivalentes) :

- $\forall \mathbf{x} \in \mathbb{R}^d,\ \mathbf{x}^T M \mathbf{x} \geq 0$
- toutes les valeurs propres sont non-négatives

On appelle vecteur propre $v \in \mathbb{R}^d$ et valeur propre $\lambda \in \mathbb{R}$ un vecteur et un réel tels que $Mv = \lambda v$. Toute matrice symétrique A réelle admet d vecteurs propres indépendants. A est donc diagonalisable : $A = Q^T \Lambda Q$ avec Q matrice orthogonale composée des vecteurs propres orthogonaux et unitaires ($\|q_i\|^2 = 1$ et $\forall i, j\ q_i.q_j = 0$), et Λ diagonale avec comme éléments les valeurs propres associées aux vecteurs propres.

Q 1.1 On démontre tout d'abord quelques propriétés du Laplacien que l'on utilisera à la fin de l'exercice.

Montrer que :

Q 1.1.1 $\forall \mathbf{x} \in \mathbb{R}^n,\ \mathbf{x}^T L \mathbf{x} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}(x_i - x_j)^2$

Q 1.1.2 L est symétrique et positive semi-définie.

Q 1.1.3 la plus petite valeur propre de L est 0 et le vecteur propre associé est le vecteur $e = (1, \ldots, 1)^T$ de taille n.

Q 1.2 On considère le cas de la séparation en 2 clusters A et $\bar{A}$, avec $k = 2$. L'objectif est de ramener l'écriture $min_{A \subset V} Rcut(A, \bar{A})$ à un problème d'optimisation continue.

Soit $\mathbf{x} \in \mathbb{R}^n$ tel que $x_i = \sqrt{\frac{|\bar{A}|}{|A|}}$ si $x_i \in A$, et $x_i = -\sqrt{\frac{|A|}{|\bar{A}|}}$ si $x_i \in \bar{A}$. La valeur de x_i permet donc d'indiquer si le nœud i appartient au premier ou au deuxième cluster. Le vecteur $\mathbf{x}$ permet d'encoder ainsi tous les partitionements possibles en deux clusters.
Montrer que

Q 1.2.1 $\mathbf{x}^T L\mathbf{x} = n.Rcut(A, \bar{A})$

Q 1.2.2 $\mathbf{x}^T.e = \sum_{i=1}^{n} x_i = 0$

Q 1.2.3 $\|x\|^2 = \mathbf{x}^T\mathbf{x} = n$. Comment interprétez vous cette contrainte et celle de la question précédente ?

On vient de montrer que le problème $min_{A \subset V} Rcut(A, \bar{A})$ est équivalent à $min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{x}^T L\mathbf{x}$ sous la contrainte d'un vecteur $\mathbf{x}$ encodant le partionnement : $\mathbf{x} = (\pm\alpha, \pm\alpha, \ldots)$. La constante α peut être quelconque, bien choisie elle permet d'équilibrer le partitionnement et d'obtenir exactement la mesure de coût *Rcut*. Prendre un $\alpha = 1$ consiste à faire un *Rcut* non pénalisé par la taille des partitions induites (ce qui peut mener à une partition dégénérée d'un seul nœud d'un côté et le reste du graphe de l'autre). Ce problème est NP-difficile : il s'agit de choisir de manière combinatoire la bonne affectation de chacun des nœuds dans une des deux partitions. Afin de le relaxer et de pouvoir le traiter par une optimisation continue, on va supprimer la contrainte sur le vecteur $\mathbf{x}$ tout en introduisant les deux contraintes $\mathbf{x}^T\mathbf{x} = n$ et $\mathbf{x}^T.e = 0$ pour garantir des solutions pas trop éloignées de notre optimum discret.

Q 1.3 Considérons le problème d'optimisation $min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{x}^T L\mathbf{x}$ sous la contrainte $\mathbf{x}^T\mathbf{x} = n$. Montrer que la solution à ce problème est un vecteur propre $e^{(i)}$ de L et que pour ce vecteur ${e^{(i)}}^T L e^{(i)} = n\lambda_i$ avec λ_i la valeur propre associée à $e^{(i)}$ (utiliser $\nabla_{\mathbf{x}} \mathbf{x}^T\mathbf{x} = 2\mathbf{x}$ et $\nabla_{\mathbf{x}} \mathbf{x}^T A\mathbf{x} = (A + A^T)\mathbf{x}$). En déduire la solution du problème.

Q 1.4 On considère le problème d'optimisation $min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{x}^T L\mathbf{x}$ sous les contraintes $\mathbf{x}^T\mathbf{x} = n$ et $\mathbf{x}^T e = 0$. En se servant de la question précédente, donner (sans calculs) la solution à ce problème. On notera e' ce vecteur.

Q 1.5 Comment se servir du vecteur propre e' pour construire une solution au problème de clustering pour $k = 2$?

Pour info, l'algorithme dans le cas général est le suivant avec k le nombre de clusters désiré. La démonstration est quasiment la même que dans le cas $k = 2$ mais en introduisant autant de vecteurs indicateurs d'appartenance de clusters que de clusters :

- Calculer le Laplacien L
- Calculer les k premiers vecteurs propres de L
- Soit $U \in \mathbb{R}^{n \times k}$ la matrice contenant les k vecteurs propres $u_1, \ldots, u_k$ en colonne
- Pour $i = 1 \ldots n$, soit $y_i \in \mathbb{R}^k$ le vecteur correspondant à la i-ème ligne de U
- Réaliser un clustering des $(y_i)_{i=1\ldots n}$ dans $\mathbb{R}^k$ avec un algorithme de k-means, on note $C_1, \ldots, C_k$ les clusters obtenus.
- les clusters de sortie sont $A_1, \ldots A_k$ avec $A_i = \{v_j \in V | y_j \in C_i\}$

Pour plus de détails, vous pouvez regarder :cet article.

Exercice 2 – Clustering et mélange de lois

On souhaite estimer une densité de probabilité par un modèle de type mélange de gaussiennes. La probabilité d'une observation x est donnée par : $p(x) = \sum_{l=1}^{L} \tau_l . p(x|\lambda_l)$ où les τ_l sont les probabilités a priori des lois et les $p(x|\lambda_l)$ sont des lois gaussiennes multi-dimentionnelles caractérisées par leur moyenne μ_l et leur matrice de co-variance Σ_l , i.e. $\lambda_l = (\mu_l, \Sigma_l)$.

Q 2.1 Dessiner la loi de probabilité pour $L = 2, \tau_1 = \tau_2 = 0.5$, et $\mu_1 = 1, \mu_2 = 3, \Sigma_1 = 1, \Sigma_2 = 10$.

Q 2.2 Quelles est la probabilité a posteriori qu'un exemple x ait été produit par la gaussienne multi-dimentionnelle l, $p(\lambda_l|x)$?

Q 2.3 Expliquer comment l'apprentissage d'un mélange de lois peut être utilisé pour faire du clustering.

Exercice 3 – Apprentisage d'un mélange de lois et maximum de vraissemblance

On souhaite apprendre le modèle de l'exercice précedent avec un critère de maximum de vraisemblance (MV) sur une base d'apprentissage $X = \{x_i\}, i = 1..N$.

Q 3.1 Exprimer la log-vraisemblance des données par le modèle en supposant que les x_i sont indépendants.

Q 3.2 Quelle est la difficulté avec cette log-vraissemblance ?

Q 3.3 L'idée de l'algorithme EM (Expectation-Maximization) est de se dire que si l'on avait des informations supplémentaires Z, on pourrait optimiser cette vraissemblance plus facilement. Quelles informations seraient utiles ici ? Donner la vraissemblance complétée par ces informations.

Q 3.4 On peut alors utiliser un algorithme dit algorithme EM (Expectation-Maximization) pour l'estimation de ce mélange de gaussiennes. Une variante de l'algorithme EM est la suivante :

- initialiser les paramètres $(\tau_i, \mu_i, \sigma_i)_{i=1..L}$;
- Répéter :
  - ▶ déterminer pour chaque x_i la gaussienne qui l'a produit avec la plus grande vraissemblance : pour $i = 1..N$, $I(x_i) = argmax_l\ p(\lambda_l|x_i)$;
  - ▶ ré-estimer les paramètres des lois à partir des exemples qui lui ont été affectés : pour $l = 1..L$, ré-estimer λ_l à partir des $\{x_i \in E|I(x_i) = l\}$

Dans le cas où tous les τ_i sont égaux (equi-probabilité des gaussiennes) et où les matrices de covariance des lois sont fixées à l'identité, montrer que l'algorithme précédent est équivalent à un algorithme des K-Moyennes.

Q 3.5 L'algorithme précédent procède par affectations successives des éléments aux différents clusters. Quelle est la limite de ce genre d'approche ?

Q 3.6 La version classique de l'algorithme EM travaille en deux étapes :

- Expectation step (E step) : Calcul de l'espérance de la log-vraissemblance en fonction des probabilités conditionnelles des données manquantes Z étant donné les observations X selon les estimations courantes des paramètres $\theta^{(t)}$:

$$Q(\theta|\theta^{(t)}) = \mathrm{E}_{Z|X,\theta^{(t)}}\left[\log L(\theta; X; Z)\right]$$

- Maximization step (M step) : Recherche des paramètres θ qui maximisent cette quantité :

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta|\theta^{(t)})$$

Q 3.7 Sachant que d'après l'inégalité de Gibbs, $\sum_{i=1}^{n} p_i \log p_i \geq \sum_{i=1}^{n} p_i \log q_i$ pour toutes paires de distributions de probabilités p et q, montrer que la suite des vraissemblances $p(X|\theta^{(t)})$, selon les paramètres $\theta^{(t)}$ calculés à chaque étape de l'algorithme EM, est croissante.

Q 3.8 Donner la formulation de l'espérance $Q(\theta|\theta^{(t)})$ selon les paramètres courants $\theta^{(t)}$.

Q 3.9 Donner alors les formulations des estimations des paramètres à l'itération t+1 selon les estimations à l'itération t