Master de Chemoinformatique et Modélisation M1S1
Examen de mathématiques pour la chimie*

MARCOU Gilles

2013-2014

Résumé

Documents autorisés. Durée : 2h. La copie de l'étudiant sera constituée d'un support papier traditionnel et d'un ou plusieurs fichiers de calcul pour Maple. ***Toute réponse doit être motivée ou sera considérée comme nulle.***

Exercice 1

Le but de cet exercice est de tester une méthode de réduction de dimensionnalité projective basée sur la description d'un jeu de donnée comme un graphe : l'algorithme de projection du Laplacien [Neural Computation, June 2003 ; 15 (6) :1373-1396]. Cette méthode de projection fait l'hypothèse que la dimensionnalité de l'espace vectoriel de plongement du jeu de donnée est trop grande : le jeu de donnée lui-même est approximativement contenu sur une variété bien plus simple. L'objectif est de projeter le jeu de données initial dans un espace de plus basse dimension (en général 2D), tout en gardant une représentation significative.

Pour commencer, chaque point du jeu de données est considéré comme un nœud d'un graphe G. Les côtés du graphe connectent deux points voisins : deux exemples i et j du jeu de données sont connectés si leur distance $d_{ij} < \epsilon$. Puis, chaque côté est pondéré et le poids d'un côté entre les nœuds i et j est stocké dans une matrice W, et la composante correspondante $W_{ij} = e^{-\frac{d_{ij}}{\sigma^2}}$ tandis que tous les éléments hors diagonaux sont nuls. Ensuite, la matrice de degré D du graphe pondéré est calculée. La matrice de degré est une matrice diagonale dont les éléments diagonaux sont la somme des composants de la ligne correspondante de la matrice W : $D_{ii} = \sum_{j=1}^{N} W_{ij}$. La matrice Laplacienne $L = D - W$ est alors évaluée. Cette matrice est interprétée comme une estimation de l'opérateur différentiel de Laplace Beltrami sur la variété échantillonnée par le jeu de données. Finalement, les vecteurs propres solutions du problème aux valeurs propres généralisé $L.v = \lambda.D.v$ ayant les plus petites valeurs propres contiennent les projections de chaque exemple sur un sous-espace pertinent, capturant des informations sur la courbure locale de la variété. Les valeurs propres sont interprétées comme une mesure de la qualité générale de la projection : des voisins de l'espace initial devant idéalement rester voisins dans la projection.

L'algorithme se décompose en trois étapes :

1. construire la matrice d'adjacence du graphe représentant les données ;
2. Pondérer les côtés est calculer les matrices W, D et L (respectivement les matrices de poids, de degré et laplacienne) ;
3. Résoudre le problème aux valeurs propres généralisé $L.v = \lambda.D.v$.

Le graphique représentant une composante principale en fonction de l'autre fournit une représentation pertinente du jeu de données initial.

*Enseignants: G. Marcou, *Université de Strasbourg, Faculté de Chimie, 1, rue Blaise Pascal, 67000 Strasbourg*

Question 1.1

Définissez les concepts suivants :
- diagonal matrice ;
- valeur propre et vecteur propre ;
- dimension d'un espace vectoriel.

La méthode est appliquée à un jeu de données artificiel de $N = 250$ points. Les points sont répartis aléatoirement entre deux variétés selon une variable aléatoire v. La position sur une variété de ces points dépend de deux autres variables aléatoires θ et ϕ. Les variables aléatoires sont échantillonnées pour générer 3 séries de valeurs pour les coordonnées x, y et z de chaque point dans un repère orthonormé. Les séries sont stockées dans 3 vecteurs N-dimensionnels.

Question 1.2

Commentez les lignes de code suivantes :

```
r1 := .5; -1; N := 250;
theta := Sample(RandomVariable(Uniform(0, 2*Pi)));
phi := Sample(RandomVariable(Uniform(0, Pi)));
v := Sample(1+RandomVariable(Bernoulli(.5)));
ltheta := theta(N); lphi := phi(N); lv := v(N);
x:=Vector([seq(r1*lv[i]*cos(ltheta[i])*sin(lphi[i]),i=1..N)]);
y:=Vector([seq(r1*lv[i]*sin(ltheta[i])*sin(lphi[i]),i=1..N)]);
z:=Vector([seq(r1*lv[i]*cos(lphi[i]),i=1..N)]);
```

Pour aider l'analyse, la variété à laquelle appartient un point est codé par une couleur, rouge ou bleu, qui est stockée dans la liste `clr`.

```
clr := [seq(COLOR(RGB, lv[i]-1, 0., 2-lv[i]), i = 1 .. N)];
```

Question 1.3

Expliquez les concepts de Maple suivant :
- une séquence
- une liste
- un ensemble

Question 1.4

Complétez la commande suivante pour tracer un graphique en 3D du nuage de points du jeu de données. Décrivez les deux variétés.

```
plots[pointplot3d](,,,color=);
```

Quelques fonctions sont fournies pour aider à réaliser l'algorithme. Par exemple, la fonction `dist2` utilise les trois vecteurs de coordonnées x, y et z et les indices i et j pour calculer le carré de la distance euclidienne entre les points i et j. La fonction `dk` mime le comportement de la fonction delta de Kronecker.

```
dist2:=
proc(x::Vector,y::Vector,z::Vector,i::integer,j::integer)::float;
return evalf((x[i]-x[j])^2+(y[i]-y[j])^2+(z[i]-z[j])^2)
end proc;
```

```
dk:=(i,j)->eval(evalb(i=j),[true=1,false=0]);

g:=(x,epsilon)->if␣epsilon␣<␣x␣then␣x␣else␣0␣end␣if;
```

Question 1.5

Expliquez ce que font, en pratique, les fonctions `g` et `dk` ?

La matrice de poids W est alors calculée avec les paramètres suivants : la largeur des gaussiennes est fixée à σ^2 (`sigma2`) et deux points sont connectés dans le graphe si la distance carré entre eux est plus petite que ϵ^2 (`epsilon2`). Pour exemple, les valeurs de ces paramètres sont $\sigma^2 = 0.1$ et $\epsilon^2 = 0.4$. Les commandes suivantes remplissent la matrice W :

```
W := Matrix([seq([seq(g(exp(-dist2(x, y, z, i, j)/sigma2)*
  (1-dk(i, j)), exp(-epsilon2/sigma2)), i = 1 .. N)],
  j = 1 .. N)]);
```

Question 1.6

Expliquez comment fonctionne la ligne calculant la matrice W.

La matrice de degré D contiens sur la diagonale la somme des composants de la ligne correspondante de la matrice W : $D_{ii} = \sum_{j=1}^{N} W_{ij}$. La matrice de degré est contenu dans la variable `De`.

Question 1.7

Complétez la commande suivante qui utilise la commande `sum` de la librairie `MTM` pour calculer la matrice de degré.

```
De␣:=␣Matrix(MTM[sum](,),shape=)
```

Question 1.8

Calculez la matrice Laplacienne $L = D - W$.

Question 1.9

Résoudre le problème de valeur propre généralisée $L.v = \lambda.D.v$. Stockez les valeurs propres dans une vecteur nommé `evl` et les vecteurs propres dans une vecteur nommé `evc`.

Question 1.10

Utilisez la commande `pointplot` de la librairie `plots`, afin de tracer une paire de vecteurs propres, la plus pertinente selon vous : ces vecteurs sont ceux qui ont la plus petite valeur propre à l'exception de celui dont la valeur propre est nulle.

Question 1.11

Livrez vos conclusions.

Barème

- Question 1.1 : 2 points
- Question 1.2 : 2 points
- Question 1.3 : 2 points
- Question 1.4 : 2 points
- Question 1.5 : 2 points
- Question 1.6 : 2 points
- Question 1.7 : 2 points
- Question 1.8 : 2 points
- Question 1.9 : 2 points
- Question 1.10 : 2 points