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

MARCOU Gilles

2014-2015

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

Cette exercice illustre la régularisation LASSO[2] utilisant l'algorithme d'optimisation SHOOT[1, 3] pour la régression multilinéaire. Dans la suite, on suivra les conventions suivantes :

- le nombre d'exemples connus pour construire un modèle est noté N_{tr} ;
- le nombre de variables explicatives est noté N_d ;
- la matrice contenant les valeurs des variables explicatives pour chacun des exemples connus est noté X_{tr} (N_{tr} lignes et N_d colonnes) ;
- le vecteur contenant la valeur de la variable expliquée de chaque exemple connu est noté Y_{tr} ; (N_{tr} éléments) ;
- le vecteur de dimension N_d qui contient les valeurs des paramètres du modèle linéaire est noté W (N_d éléments).

La régulariasation LASSO vise à déterminer les paramètres optimum $\hat{W}$ minimisant la fonction objectif suivante (qu'on appelera aussi LASSO) :

$$\|Y_{tr} - X_{tr}.W\|^2 + 2\lambda|W| \tag{1}$$

où la notation $\|.\|$ désigne la norme Euclidienne usuelle et la notation $|.|$ désigne la norme L_1 qui consiste, pour un vecteur, à faire la somme des valeurs absolues de toutes les composantes d'un vecteur. Le paramètre λ contrôle l'importance relative du la norme de vecteur W par rapport à la qualité de l'ajustement du modèle sur les données ; c'est un paramètre de l'algorithme qui sera fixé à la valeur 4 pour cet exercice.

Question 1.1

Expliquez en quelques mots les notions suivante :

- la norme d'un vecteur,
- un espace Euclidien,
- une différentielle de fonction,
- dérivée de fonction.

Les données sont chargées en mémoire :

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

```
f := fopen("Xtr.dat", READ, TEXT): Xtr := Matrix(readdata(f,
                    float, Nd)): fclose(f):
f := fopen("Ytr.dat", READ, TEXT): Ytr := Vector(readdata(f,
                    float), orientation = column): fclose(f):
```

Ensuite les variables explicatives et la variable expliquée sont centrés. Le résultat de cette opération est que la moyenne des composantes de Y_{tr} est nulle et que la moyenne des composantes de la matrice X_{tr} prises par colonne est nulle également.

```
Ymean := Statistics[Mean](Ytr):
Ymean := Vector(1 .. Ntr, fill = Ymean):
Xmean := [seq(Statistics[Mean](Xtr[1 .. Ntr, i]), i = 1 .. Nd)]:
Xmean := Matrix([seq(Xmean, j = 1 .. Ntr)]):
Xtr := Xtr-Xmean:
Ytr := Ytr-Ymean:
```

Question 1.2

Expliquez le fonctionnement des 6 lignes ci-dessus. Vous vous attacherez à identifier quand une variable désigne un nombre, une séquence, une liste, un vecteur ou une matrice.

Le principe de l'algorithme SHOOT est d'étudier la dérivée de la fonction objectif relativement au paramètres W, quand aucune composante du vecteur n'est nulle. Puis, l'algorithme cherche à annuler chaque composante de la dérivée tour à tour. Comme chaque équation à résoudre pour une composante dépend des valeurs prises par les autres composantes, l'ensemble de la procédure est répétée pour approcher le minimum.

L'algorithme nécessite donc dans un premier temps, une phase préparatoire où certains calculs sont effectués une fois pour toute afin d'en réutiliser les résultats ultérieurement. Ces calculs incluent :

- la matrice XX résultant du produit matriciel de X_{tr} avec sa propre transposée X_{tr}^T : $XX = X_{tr}^T.X_{tr}$;
- le vecteur XY résultant du produit de la matrice X_{tr} sur le vecteur Y_{tr}.
- un vecteur $\mathrm{diag}(XX)$ donc chaque composante est l'élément diagonal correspondant de la matrice XX (la commande `Diagonal` de la librairie `LinearAlgebra` est particulièrement recommandée) ;
- la matrice XX_0 dont tous les éléments diagonaux sont nuls et les éléments hors diagonaux sont égaux à ceux de la matrice XX.

Question 1.3

Calculez les éléments suivants :

- la matrice XX qui sera stockée dans une variable du même nom, `XX` ;
- le vecteur XY qui sera stocké dans une variable homonyme, `XY` ;
- le vecteur $\mathrm{diag}(XX)$ qui sera stocké dans une variable appelée `XXd` ;
- la matrice XX_0 qui sera stockée dans une variable appelée `XX0`.

Il est ensuite nécessaire d'initialiser le vecteur de paramètre W afin qu'il ne soit pas trop éloigné du minimum recherché afin d'accélérer la convergence de l'algorithme. La solution retenue est d'utiliser la solution de la régression Ridge pour cela. La régression Ridge utilise une régularisation de Tikhonov, ce qui se traduit pas la minimisation de l'expression suivante, avec les même notations :

$$\|Y_{tr} - X_{tr}.W\|^2 + \lambda\|W\|^2 \tag{2}$$

La solution est alors très simple :

$$W = (X_{tr}^T.X_{tr} + \lambda \mathbf{1})^{-1}.X_{tr}^T.Y \tag{3}$$

Avec la matrice unité notée $\mathbf{1}$. Dans ce calcul on identifie le paramètre λ de l'algorithme Ridge avec la paramètre λ de la fonction LASSO. En pratique cela conduit à initialiser W comme suit :

```
W := Multiply(MatrixInverse(XX+lambda*IdentityMatrix(Nd)), XY):
```

Question 1.4

Expliquez avec vos mots comment fonctionne la commande ci-dessus.

Dans l'algorithme SHOOT, l'anulation d'une composante j du gradiant de la fonction LASSO conduit à une forme du paramètre correspondant W_j représentée par la fonction suivante :

```
g := (x, y) -> piecewise(x < -lambda, (-lambda-x)/y,
          x <= lambda, 0, lambda < x, (lambda-x)/y):
```

Question 1.5

— Tracer la fonction $g(x, y)$ ci-dessus pour des valeurs de $x \in [-10, 10]$ et $y = 1$.
— Expliquez ce que fait la fonction `piecewise`.

Notez que quelque soit la valeur de y il existe des situations (des valeurs de x) pour lesquels le paramètre W_j va s'annuler. C'est une caractéristique importante de ce type de méthodes : elles agissent comme une sélection de variable, produisant au final des modèles plus simples à mettre en oeuvre et à intepréter.

Pour annuler la composante j du grandiant, l'algorithme SHOOT propose de calculer une quantité

$$S = -XY_j + [XX_0.W]_j \tag{4}$$

où $[XX[0].W]_j$ représente la composante j du produit de la matrice XX_0 sur le vecteur W. La nouvelle valeur pour le paramètre j est alors :

$$w_j = \begin{cases} \frac{-\lambda - S}{\text{diag}(XX)_j}, \text{si } S < -\lambda \\ 0, \text{si } -\lambda \leq S \leq \lambda \\ \frac{\lambda - S}{\text{diag}(XX)_j}, \text{si } S > \lambda \end{cases} \tag{5}$$

en d'autres termes $W_j = g(S, \text{diag}(XX)_j)$. Pour chaque composante j, on calcul la quantité S puis on met à jour la composante W_j correspondante.

La procédure est répétée jusqu'à convergence du vecteur de paramètre W. Dans cet exercice on vérifie que la norme Euclidienne du vecteur W ne varie plus beaucoup.

Question 1.6

Complétez les boucles `for` intriquées ci-dessous pour calculer S et mettre à jour les composantes du vecteur W pour chacune des N_d composantes et répéter jusqu'à convergence. Les éléments manquant sont identifiés par des **?**.

```
Normold := 0:
for m while abs(Normold-Norm(W, ?)) > 0.1e-2 do
  Normold := Norm(W, ?);
  for n to ? do
    S := -?[n]+?[n];
    W[n] := g(?, ?)
  end do:
  m, Norm(W, ?), abs(Normold-Norm(W, ?));
end do;
```

Le modèle peut ensuite être utilisé pour calculer les valeurs estimées de la variable expliquée : $Y_{pred} = X_{tr}.W$.

Question 1.7

- Calculez les valeurs estimées Y_{pred} et stockez les dans la variable `Ypred`.
- Tracez les valeurs estimées Y_{pred} en fonction des valeurs réelles Y_{tr}.
- Tracez un histogramme pour représenter les valeurs des composantes du vecteur W. Cherchez la fonction adaptée dans la librairie Statistics.

Question 1.8

Donnez ici quelques phrases de conclusion.

Exercice 1

Dans cette exercice on se propose d'étudier une réaction chimique suivant une cinétique d'ordre 2, du type $A + B \rightarrow C$. Si la concentration initiale du composé A est notée a et la concentration initiale du composé B est notée b, alors on suit la réaction en comptabilisant la quantité de produit $x(t)$ consommée à un instant donné. Cela conduit à l'équation différentielle :

$$\frac{dx(t)}{dt} = -k.(a - x(t)).(b - x(t)) \tag{6}$$

Question 2.1

- Stocker dans une variable l'équation différentielle ci-dessus.
- Définir la condition initiale $x(0)$.
- Résoudre l'équation formelle.
- Tracer la solution pour $k = 0.025$, $a = 20$, et $b = 10$ pendant une durée de 0 à 1 unité de temps.

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 2.1 : 5 points

Références

[1] WJ Fu. Penalized regressions : The bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7(3) :397–416, 1998.

[2] T Hastie, R Tibshirani, and JH Friedman. *The Elements of Statistical Learning*. Springer, 2008.

[3] M Schmidt. Least squares optimization with l1-norm regularization, 2005.