Optimisation à finalité statistique

Salim Lardjane

Université de Bretagne-Sud

Cours 8 - Applications Statistiques :
Données incomplètes et algorithme EM

Données incomplètes

De nombreux problèmes d'estimation font intervenir des *données manquantes*. Une approche générale de ces problèmes d'estimation peut être développée en associant la méthode du maximum de vraisemblance avec un algorithme particulier, appelé *algorithme EM*.

Exemple classique

Considérons par exemple une variable multinômiale $x = (x_1, x_2, x_3, x_4, x_5)'$ de loi

$$\mathcal{M}(N, p_1, p_2, p_3, p_4, p_5)$$

où les p_i dépendent d'un paramètre π de la façon suivante

$$\begin{aligned} p_1 &= 1/2 \\ p_2 &= \pi/4 \\ p_3 &= \frac{1-\pi}{4} \\ p_4 &= \frac{1-\pi}{4} \\ p_5 &= \pi/4 \end{aligned}$$

Des modèles tels que celui-ci sont notamment utilisés en génétique des populations.

Exemple classique

Etant donnée une observation

$$(x_1, x_2, x_3, x_4, x_5)'$$

on peut déterminer l'estimation de π par maximum de vraisemblance en écrivant

$$\mathcal{L}(\pi) \propto \left(\frac{1}{2}\right)^{x_1} \left(\frac{\pi}{4}\right)^{x_2} \left(\frac{1-\pi}{4}\right)^{x_3} \left(\frac{1-\pi}{4}\right)^{x_4} \left(\frac{\pi}{4}\right)^{x_5}$$

avec un coefficient de proportionnalité égal à

$$\frac{N!}{x_1!x_2!x_3!x_4!x_5!}$$

Exemple classique

Par conséquent, la log-vraisemblance s'écrit

$$\begin{aligned}\ell(\pi) &= Cste + x_1 \log(\frac{1}{2}) + x_2 \log(\frac{\pi}{4}) \\ &\quad + x_3 \log(\frac{1-\pi}{4}) \\ &\quad + x_4 \log(\frac{1-\pi}{4}) \\ &\quad + x_5 \log(\frac{\pi}{4})\end{aligned}$$

d'où

$$4\,\frac{d\ell}{d\pi}(\pi) = \frac{x_2 + x_5}{\pi} - \frac{x_3 + x_4}{1-\pi}$$

Exemple classique

On en déduit l'estimation de π par maximum de vraisemblance

$$\widehat{\pi} = \frac{x_2 + x_5}{x_2 + x_3 + x_4 + x_5}.$$

Supposons à présent, qu'au lieu de

$$(x_1, x_2, x_3, x_4, x_5)'$$

on observe $(y_1, y_2, y_3, y_4)'$ *où*

$$\begin{aligned} y_1 &= x_1 + x_2 \\ y_2 &= x_3 \\ y_3 &= x_4 \\ y_4 &= x_5 \end{aligned}$$

Exemple classique

On ne dispose plus des *données complètes* permettant d'estimer π par maximum de vraisemblance direct.

Est-il néanmoins possible d'estimer π ?

Oui. Cela peut être fait à l'aide de l'algorithme suivant :

Exemple classique

1. Etant donnée une valeur courante de π, notons-la π^i, initialisée arbitrairement, utiliser les données observées pour *estimer les données complètes* $(x_1, x_2, x_3, x_4, x_5)'$.
   Comme x_3, x_4 et x_5 sont connues, on n'a besoin d'estimer que x_1 et x_2. Cela peut être fait en posant

   $$\widehat{x}_1 = y_1 \frac{1/2}{1/2 + \pi^i/4} \quad \widehat{x}_2 = y_1 \frac{\pi^i/4}{1/2 + \pi^i/4}$$

   (admis).

2. Utiliser les données complètes estimées pour obtenir une nouvelle estimation π^{i+1} de π

3. Alterner les étapes 1) et 2) jusqu'à ce qu'un critère de convergence soit vérifié.

Exemple classique

Exercice 9. Implémenter l'algorithme ci-dessus pour les données incomplètes $y_1 = 125$, $y_2 = 18$, $y_3 = 20$, $y_4 = 34$ et la valeur initiale $\pi^0 = 0.5$.
Adopter comme critère de convergence de l'algorithme

$$|\pi^{i+1} - \pi^i| < \varepsilon = 10^{-7}$$

Mélange de deux gaussiennes

Intéressons-nous à présent à l'estimation des cinq paramètres du mélange de deux lois normales univariées.

La fonction de densité correspondante est de la forme

$$f(x) = pf_1(x) + (1-p)f_2(x)$$

avec

$$f_i(x) = \frac{1}{\sqrt{2\pi}\,\sigma_i} \exp\left\{-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right\} \quad (i = 1, 2)$$

Mélange de deux gaussiennes

Etant donné un échantillon $x_1, \ldots, x_n$ issu de la loi de mélange, la log-vraisemblance s'écrit

$$\ell(\mu_1, \mu_2, \sigma_1, \sigma_2, p) = \sum_i \log(p f_1(x_i) + (1-p) f_2(x_i))$$

En dérivant cette expression par rapport à chacun des paramètres, en égalant les dérivées obtenues à zéro et moyennant quelques calculs relativement directs, on obtient

Mélange de deux gaussiennes

$$
\begin{aligned}
\widehat{p} &= \frac{1}{n}\sum_i \widehat{P}(C_1|x_i) \\
\widehat{\mu}_j &= \frac{\sum_i \widehat{P}(C_j|x_i)\, x_i}{\sum_i \widehat{P}(C_j|x_i)} \quad (j=1,2) \\
\widehat{\sigma_j}^2 &= \frac{\sum_i \widehat{P}(C_j|x_i)(x_i-\widehat{\mu}_j)^2}{\sum_i \widehat{P}(C_j|x_i)} \quad (j=1,2)
\end{aligned}
$$

où l'on a noté

$$\widehat{P}(C_1|x_i) = \widehat{p}\,\frac{f_1(x_i)}{f(x_i)}, \quad \widehat{P}(C_2|x_i) = 1-\widehat{P}(C_1|x_i)$$

les *probabilités a posteriori estimées* d'appartenance aux composantes du mélange.

Mélange de deux gaussiennes

Dans ce cas également, on peut considérer que *les données sont incomplètes* en ce sens que l'on ne sait pas à quelle composante parmi les deux composantes du mélange est associée chaque observation x_i.

Cependant, grâce à un algorithme analogue à celui décrit dans l'exemple précédent, on peut déterminer les estimations requises de la façon suivante :

Mélange de deux gaussiennes

1. Etant données les valeurs courantes de p, μ_1, μ_2, σ_1, σ_2, utiliser les données observées pour estimer les probabilités a posteriori $\widehat{P}(C_1|x_i)$, $\widehat{P}(C_2|x_i)$.

2. Injecter les probabilités a posteriori estimées dans le système d'équations précédent de façon à obtenir de nouvelles estimations des paramètres.

3. Alterner les étapes 1) et 2) jusqu'à ce qu'un critère de convergence adapté soit vérifié.

Mélange de deux gaussiennes

Exercice 10. Générer cinquante observations selon la loi de mélange

$$f(x) = pf_1(x) + (1-p)f_2(x)$$

avec $p = 0.4$, $\mu_1 = 0$, $\mu_2 = 3$, $\sigma_1^2 = 0.5$, $\sigma_2^2 = 1$.

Implémenter ensuite l'algorithme précédent en partant des valeurs initiales $p^0 = 0.2$, $\mu_1^0 = 1$, $\mu_2^0 = 2$, $(\sigma_1^0)^2 = 1$, $(\sigma_2^0)^2 = 0.5$.

Mélange de deux gaussiennes

On adoptera comme critère de convergence que la distance euclidienne entre deux estimations successives des paramètres soit inférieure à $\varepsilon = 10^{-4}$.

On représentera la densité réelle et la densité estimée sur un même graphique.

On résoudra également le problème à l'aide des algorithmes du simplexe de Nelder-Mead, de quasi-Newton BFGS, des gradients conjugués et de Newton-Raphson avec évaluation des gradient et hessienne par différences finies (implémentation par défaut sous R avec la fonction **optim()**).

On comparera les résultats obtenus à ceux obtenus par l'algorithme précédent (algorithme EM).