Optimisation à finalité statistique

Salim Lardjane

Université de Bretagne-Sud

Cours 2 - Algorithmes de minimisation

Algorithmes de minimisation

Les techniques de minimisation qui seront présentées dans la suite ont toutes certaines *caractéristiques en commun*.

Tout d'abord, elles sont *itératives* (récursives) et elles procèdent en général en générant *une suite* $\{\theta^i\}$ de solutions approchées dont chacune fournit une meilleure valeur approchée des paramètres au minimum de f que les précédentes, c'est-à-dire qu'on a

$$f(\theta^{i+1}) \leq f(\theta^i)$$

où θ^i et θ^{i+1} sont des vecteurs correspondant aux valeurs approchées des paramètres aux étapes i et $i+1$ respectivement.

Algorithmes de minimisation

Ainsi, peut résumer le processus de minimisation par le schéma suivant :

$$
\begin{array}{ccccccccc}
\theta^0 & \rightarrow & \cdots & \rightarrow & \theta^i & \rightarrow & \theta^{i+1} & \rightarrow & \cdots \\
f(\theta^0) & \geq & \cdots & \geq & f(\theta^i) & \geq & f(\theta^{i+1}) & \geq & \cdots
\end{array}
$$

Algorithmes de minimisation

Ces procédures (ou algorithmes) requièrent en général que l'utilisateur spécifie une valeur/estimation initiale θ^0 à partir de laquelle les approximations successives sont obtenues via une *formule de mise à jour* de la forme

$$\theta^{i+1} = \theta^i + h_i \, d_i$$

ce qui est parfois noté succinctement

$$\theta \leftarrow \theta + h \, d$$

Dans cette formule de mise à jour, d_i est un vecteur de dimension m spécifiant la *direction* du déplacement à effectuer pour passer de θ^i à θ^{i+1} et h_i est un scalaire strictement positif spécifiant la *distance* sur laquelle on se déplace dans la direction d.

Algorithmes de minimisation

Le choix d'une direction et d'une distance (ou *pas* du déplacement) adpatées est fait de façon à ce que la condition $f(\theta^{i+1}) \leq f(\theta^i)$ soit vérifiée. Cela peut être fait de diverses manières.

(i) il peut être basé uniquement sur les valeurs de la *fonction à minimiser* en plus des informations obtenues aux itérations précédentes ; on parle alors d'*algorithmes de recherche directe*

(ii) il peut être basé également sur les dérivées partielles de f par rapport aux paramètres ; on parle alors d'*algorithmes de gradient*.

Algorithmes de minimisation

Une question commune à l'ensemble de ces algorithmes est celle du *critère d'arrêt* de la procédure.

En général, ce critère est basé sur les suites $\{\theta^i\}$ et $\{f(\theta^i)\}$. Des critères d'arrêt possibles sont par exemple

1) $|f(\theta^{i+1}) - f(\theta^i)| < \varepsilon$

2) $||\theta^{i+1} - \theta^i|| < \varepsilon'$

où ε et ε' sont des constantes fixées.

Algorithmes de minimisation

Bien que ces critères soient utilisés en pratique, et dans bien des cas de façon satisfaisante, ils peuvent avoir pour effet d'arrêter l'algorithme d'optimisation *prématurément*.

Pour éviter ce genre de situation, on peut retenir un critère d'arrêt plus contraignant, en requérant que 1) et 2) soient vérifiées pour plusieurs itérations successives.

Algorithmes de minimisation

Notons que les critères d'arrêt comme 1) et 2) dépendent fortement de l'*échelle* de la fonction f et des paramètres $\theta_1, \ldots, \theta_m$.

Par exemple, si $\varepsilon = 10^{-3}$ et f prend ses valeurs dans l'intervalle $[10^{-7}, 10^{-5}]$, alors n'importe quelle valeur de θ vérifie la condition 1).

De même, le critère 2) peut poser problème si les échelles des paramètres sont très différentes.

Par exemple, si $m = 2$, $\theta_1 \in [10, 100]$ et $\theta_2 \in [0.001, 0.01]$, alors le critère 2) ignorera pratiquement le second paramètre.

Algorithmes de minimisation

Ce problème d'échelle affecte les méthodes d'optimisation qui ne sont pas invariantes par changement d'échelle.

La solution consiste à choisir des unités pour les paramètres de façon à ce qu'ils soient de magnitude comparable.