Estimation de choix discrets avec l'algorithme du gradient stochastique

Alexandre Brilhante

IFT3150: Projet d'informatique
Département d'informatique et de recherche opérationnelle
Université de Montréal

alexandre.brilhante@umontreal.ca

5 décembre 2017

Table des matières

Illustration

Où allons-nous dîner?

Plusieurs paramètres possibles peuvent influencer ce choix:

- Budget
- Temps disponible
- Distance
- Goûts personnels
- etc.

Contexte

Un agent économique fait face à un choix, ou une série de choix au cours du temps, parmi un ensemble fini d'alternatives. Nous voulons observer le processus comportemental menant au choix de l'agent.

L'agent veut maximiser son utilité. On suppose donc que nous pouvons capturer les préférences individuelles au moyen d'utilité, connue par l'agent mais pas par le chercheur, associée à chaque alternative.

Puisque les utilités sont aléatoires, nous pouvons seulement calculer des probabilités de choix.

Ensemble de choix

Les alternatives forment l'ensemble de choix.

Propriétés:

- L'ensemble de choix doit être exhaustif et donc représenter toutes les alternatives possibles.
- Les alternatives doivent être mutuellement exclusives: choisir une alternative fait en sorte que toutes les autres sont rejetées.
- L'ensemble de choix doit contenir un nombre fini d'alternatives.

Fonction de choix

Pour tenir compte des attributs et caractéristiques non-observées, nous décomposons les utilités comme suit:

$$U_{ij} = V_{ij} + \epsilon_{ij}$$

où:

$$V_{ij} = V_{ij}(\beta_j, x_{ij}) = \beta_j^T x_{ij} = \sum_{k=1}^{K_j} \beta_j^k x_{ij}^k$$

et:

- i : individu
- j : alternatives
- x : vecteur contenant tous les attributs
- K_j : nombre d'attributs pour l'alternative
- β : vecteur de paramètres à estimer

Probabilité de choix

Fonction de choix:

$$U_{ij} = V_{ij} + \epsilon_{ij}$$

Assumons que les ϵ_{ij} soient indépendants et identiquement distribués, nous avons donc:

$$U_{ij} = V_{ij}$$

Dès lors, la probabilité de choix est:

$$P_{ij}(\beta) = \frac{e^{\beta V_{ij}}}{\sum_{j=1}^{|A(j)|} e^{\beta V_{ij}}}$$

Logit

Mixed Logit

Estimation

Newton multidimensionnel

Newton multidimensionnel: limites

La convergence n'est assurée que si le point de départ est suffisamment proche de la solution, et est alors quadratique.

On peut obtenir la convergence globale en utilisant la direction de Newton comme direction de descente avec une méthode de recherche linéaire, en s'assurant que ak puisse prendre la valeur 1 afin d'assurer une convergence quadratique au cours des dernières itérations.

Régions de confiance

Nous considérons le problème:

$$\min_{x \in \Re^n} f(x)$$

où $f(x) \in C^1$ et $\nabla f(x)$ est lipschitzienne sur $\Re^n$

Les algorithmes d'optimisation itératifs travaillent typiquement à chaque itération avec un problème d'optimisation bien plus simple.

- Dans le cas des méthodes de recherche linéaire, le sous-problème est plus simple car il est unidimensionnel.
- En régions de confiance, le sous-problème reste en dimension n mais la fonction objectif est simplifiée et l'espace de recherche restreint.

Régions de confiance

Chaque sous-problème de région de confiance doit être résolu approximativement et aussi efficacement que possible.

Afin d'être capable de garantir la convergence globale de la méthode, nous voudrions au minimum que la solution approximative atteigne le même niveau de réduction de la valeur du modèle que ne le ferait un pas dans la direction de plus forte pente, contraint par la région de confiance.

Étapes:

- Choisir un pas s_k pour réduire le modèle de $f(x_k + s)$
- Accepter $x_{k+1} = x_k + s_k$ si la réduction prédite par le modèle est vérifiée en $f(x_k + s_k)$
- Autrement, poser $x_{k+1} = x_k$ et raffiner le modèle

Régions de confiance avec gradient conjugué

Calcul du pas: gradient conjugué.

Régions de confiance avec gradient conjugué tronqué

Calcul du pas: gradient conjugué tronqué, aussi connu sous le nom de méthode de Steihaug-Toint.

Résultats partiels

Algorithme du gradient stochastique

Pseudocode

Input: initial vector of parameters ω and learning rate η

while an approximate minimum is not obtained **do**

  Randomly shuffle examples in the training set

  for $i = 1, 2, 3, ..., n$ **do**

    $\omega = \omega - \eta \nabla Q_i(\omega)$

  end for

end while

Ce qu'il reste à faire

- Implémenter l'algorithme du gradient stochastique en Julia.
- Estimer des jeux de données plus gros.
- Comparer les résultats des différentes méthodes.
- Finaliser le rapport.

Réferences

Fabian Bastin (2004)
Trust-Region Algorithms for Nonlinear Programming and Mixed Logit Models
Presses Universitaires de Namur

Kenneth Train (2012)
Discrete Choice Methods with Simulation
Cambridge University Press

Questions? Merci!