SNM: Stochastic Newton Method for optimization of Discrete Choice Models

Gael Lederrey
Transport and Mobility Laboratory
École Polytechnique Fédérale de Lausanne
Station 18, CH-1015 Lausanne
gael.lederrey@epfl.ch

Virginie Lurkin
Transport and Mobility Laboratory
École Polytechnique Fédérale de Lausanne
Station 18, CH-1015 Lausanne
virginie.lurkin@epfl.ch

Michel Bierlaire
Transport and Mobility Laboratory
École Polytechnique Fédérale de Lausanne
Station 18, CH-1015 Lausanne
michel.bierlaire@epfl.ch

***Abstract*—BLA BLA BLA**

***Index Terms*—Discrete Choice Models, Optimization**

I. Introduction

☞ **NOTE:**
- Not a lot of work on optimization of DCMs
- ML are doing this a lot
- Expecting a lot more data. =¿ Standard quasi Newton may have troubles
- Becomes interesting to search for new algorithms exploiting the structure of DCMs

II. Related Work

☞ **NOTE:**
- ???

III. Methodology

In this section, we present the model used in this paper, the algorithms being tested as well as the Stochastic Newton Method.

A. Model

We use the *Swissmetro* dataset [1] and build a multinomial logit model denoted by $\mathcal{M}$:

$$\begin{aligned} V_{\text{Car}} &= \text{ASC}_{\text{Car}} + \beta_{\text{TT,Car}}\text{TT}_{\text{Car}} + \beta_{\text{C,Car}}\text{C}_{\text{Car}} + \beta_{\text{Senior}}\mathbb{1}_{\text{Senior}} \\ V_{\text{SM}} &= \text{ASC}_{\text{SM}} + \beta_{\text{TT,SM}}\text{TT}_{\text{SM}} + \beta_{\text{C,SM}}\text{C}_{\text{SM}} \\ &\quad + \beta_{\text{HE}}\text{HE}_{\text{SM}} + \beta_{\text{Senior}}\mathbb{1}_{\text{Senior}} \\ V_{\text{Train}} &= \text{ASC}_{\text{Train}} + \beta_{\text{TT,Train}}\text{TT}_{\text{Train}} + \beta_{\text{C,Train}}\text{C}_{\text{Train}} + \beta_{\text{HE}}\text{HE}_{\text{Train}} \end{aligned} \tag{1}$$

where $\mathbb{1}_{\text{Senior}}$ is a boolean variable equal to one if the age of the respondent is over 65 years olds, 0 otherwise, C denotes the cost, TT the travel time, and HE the headway for the train and Swissmetro. On this model, we remove all observations with unknown choice, unkown age and non-positive travel time. This gives a total of 9,036 observations.

This model is first estimated with Biogeme [2] to obtain the optimal parameter values and verify that all parameters are significant. However, we do not use the usual log-likelihood. Instead, we are using a nomralized log-likelihood which simply corresponds to the log-likelihood divided by the number of observations. Therefore, the final normalized log-likelihood is -0.7908 and the parameters are given in Table I.

Name	Value	Std err	t-test	p-value
ASC_{Car}	0	-	-	-
ASC_{SM}	$7.86 \cdot 10^{-1}$	$6.93 \cdot 10^{-2}$	11.35	0.00
$\text{ASC}_{\text{Train}}$	$9.83 \cdot 10^{-1}$	$1.31 \cdot 10^{-1}$	7.48	0.00
$\beta_{\text{TT,Car}}$	$-1.05 \cdot 10^{-2}$	$7.89 \cdot 10^{-4}$	-8.32	0.00
$\beta_{\text{TT,SM}}$	$-1.44 \cdot 10^{-2}$	$6.36 \cdot 10^{-4}$	-21.29	0.00
$\beta_{\text{TT,Train}}$	$-1.80 \cdot 10^{-2}$	$8.65 \cdot 10^{-4}$	-20.78	0.00
$\beta_{\text{C,Car}}$	$-6.56 \cdot 10^{-3}$	$7.89 \cdot 10^{-4}$	-8.32	0.00
$\beta_{\text{C,SM}}$	$-8.00 \cdot 10^{-3}$	$3.76 \cdot 10^{-4}$	-21.29	0.00
$\beta_{\text{C,Train}}$	$-1.46 \cdot 10^{-2}$	$9.65 \cdot 10^{-4}$	-15.09	0.00
β_{Senior}	-1.06	$1.16 \cdot 10^{-1}$	-9.11	0.00
β_{HE}	$-6.88 \cdot 10^{-3}$	$1.03 \cdot 10^{-3}$	-6.69	0.00

TABLE I
Parameters of the optimized model $\mathcal{M}$ by Biogeme.

We also provide a normalized model $\mathcal{M}_N$ where the values of travel time, cost and headway have been divided by 100. The parameters for this nomralized model are the same as model $\mathcal{M}$ except that the values of parameters associated to the features normalized are multiplied by 100. This is done such that all the parameters are in only one order of magnitude as opposed to the values in Table I where the parameter values are in four orders of magnitude.

B. Algorithms

To train models $\mathcal{M}$ and $\mathcal{M}_N$, many different algorithms were used. These algorithms fall in three different categories: first-order methods, second-order methods and quasi-newton methods. As first-order methods, we use mini-batch SGD [3] and Adagrad [4]. For the quasi-newton methods, we use BFGS algorithm [5] and RES-BFGS [6], a regularized stochastic version of BFGS. The main second-order algorithm is the Newton method [7]. All the algorithms presented above are run with a backtracking Line Search method using the Armijo-Goldstein condition [8] to avoid the long and tedious search of a good learning rate.

C. Stochastic Newton Algorithm

In this paper, we present an algorithm called Stochastic Newton Method. Within Neural Networks, the number of features K can easily exceed one million

> ☞ **NOTE**: REFERENCE

. Thus, this is leading to extremely large Hessian since it will have K^2 elements. Discrete Choice Models, on the other hand, tend to have a reasonable number of features. Indeed, since the main purpose of Discrete Choice Models is explaining the behavioral aspect of the samples, the models cannot contain too many parameters

> ☞ **NOTE**: REFERENCE?

. Therefore, the main limitation of Newton methods encountered in Neural Networks is not valid for Discrete Choice models. Yet one problem still remains: the exponential growth of data. Indeed, computing the Hessian on many data can be as tedious as computing it for many features. Thus the need of a Stochastic Newton Method (SNM).

The main point of this algorithm is to compute a stochastic Hessian. We show here that computing a stochastic Hessian is possible for a Logit Model. The generalization can be applied to any finite-sum function as the log-likelihood of a Logit Model. Let N denote the number of samples, $\mathcal{C}$ denote the choice set and $\mathcal{C}_n$ denote the choice set available for observation n and define

$$y_{in} = \begin{cases} 1 & \text{if observation } n \text{ chose alternative } i, \\ 0 & \text{otherwise.} \end{cases}$$

The likelihood function for a choice model is given by

$$\mathcal{L}^* = \prod_{n=1}^{N} \prod_{i \in \mathcal{C}_n} P_n(i)^{y_{in}} \tag{2}$$

where $P_n(i)$ denotes the probability that observation n choses alternative i. For a Logit model, we can define this probability as

$$P_n(i) = \frac{e^{V_{in}}}{\sum_{j \in \mathcal{C}_n} e^{V_{jn}}} \tag{3}$$

where V_{in} denotes the utility of alternative i for observation n. If we take the logarithm of equation 3, we get the log-likelihood:

$$\begin{aligned} \mathcal{L} &= \sum_{n=1}^{N} \sum_{i \in \mathcal{C}_n} y_{in} \left(V_{in} - \ln \sum_{j \in \mathcal{C}_n} e^{V_{jn}} \right) \\ &= \sum_{n=1}^{N} \left(\sum_{i \in \mathcal{C}_n} y_{in} V_{in} - \ln \sum_{j \in \mathcal{C}_n} e^{V_{jn}} \right) \end{aligned} \tag{4}$$

The second equality is done using the fact that $\sum_{i \in \mathcal{C}_n} y_{in} = 1$. We then update the log-likehood of equation 4 to create a normalized log-likelihood.

$$\bar{\mathcal{L}} = \frac{1}{N} \mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \left(\sum_{i \in \mathcal{C}_n} y_{in} V_{in} - \ln \sum_{j \in \mathcal{C}_n} e^{V_{jn}} \right) \tag{5}$$

This is done such that the value of the log-likelihood stay in the same magnitude of order for any subset of observations $\mathcal{I}$. Indeed, if we denote $\mathcal{L}_{\mathcal{I}}$ as the log-likelihood computed on the observation from $\mathcal{I}$ and $\mathcal{N}$ the set of all observations, we see that

$$\begin{aligned} \mathcal{L}_{\mathcal{I}} &= \sum_{n \in \mathcal{I}} \left(\sum_{i \in \mathcal{C}_n} y_{in} V_{in} - \ln \sum_{j \in \mathcal{C}_n} e^{V_{jn}} \right) \\ &< \sum_{n \in \mathcal{I}} \left(\sum_{i \in \mathcal{C}_n} y_{in} V_{in} - \ln \sum_{j \in \mathcal{C}_n} e^{V_{jn}} \right) \\ &\quad + \sum_{n \in \mathcal{N} \setminus \mathcal{I}} \left(\sum_{i \in \mathcal{C}_n} y_{in} V_{in} - \ln \sum_{j \in \mathcal{C}_n} e^{V_{jn}} \right) \\ &= \mathcal{L} \end{aligned} \tag{6}$$

As shown in equation 6, the standard log-likelihood cannot be compared on different set of data if they do not have the same number of data. Therefore, it can be shown that normalizing this log-likelihood as done in equation 5 produces log-likelihood of same order of magnitude independently of the number of observations.

The first derivatives of $\bar{\mathcal{L}}$ with respect to the coefficient for $k = 1, \ldots, K$ are given by

$$\begin{aligned} \frac{\partial \bar{\mathcal{L}}}{\partial \beta_k} &= \frac{1}{N} \sum_{n=1}^{N} \left(\sum_{i \in \mathcal{C}_n} y_{in} \frac{\partial V_{in}}{\partial \beta_k} - \sum_{i \in \mathcal{C}_n} 1 P_n(i) \frac{\partial V_{in}}{\partial \beta_k} \right) \\ &= \frac{1}{N} \sum_{n=1}^{N} \sum_{i \in \mathcal{C}_n} \left(y_{in} - P_n(i) \right) \frac{\partial V_{in}}{\partial \beta_k} \end{aligned} \tag{7}$$

The second derivatives for $k = 1, \ldots, K$ and $l = 1, \ldots, K$ are given by

$$\frac{\partial^2 \bar{L}}{\partial \beta_k \partial \beta_l} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{i \in \mathcal{C}_n} P_n(i) W_{ink} W_{inl} \tag{8}$$

where

$$W_{ink} = \left(\frac{\partial V_{in}}{\partial \beta_k} - \sum_{j \in \mathcal{C}_n} \frac{\partial V_{jn}}{\partial \beta_k} P_n(j) \right)$$

From the definition of the second derivatives in equation 8, it is easy to compute the second derivative for only one observation m.

$$\left. \frac{\partial^2 \bar{L}}{\partial \beta_k \partial \beta_l} \right|_m = - \sum_{i \in \mathcal{C}_m} P_m(i) W_{imk} W_{iml} \tag{9}$$

From the definitions in equations 8 and 9, we can conclude that the Hessian on a subset of the observations $\mathcal{I}$ is the average of the Hessians for each of observation $i \in \mathcal{I}$.

We present now the Stochastic Newton Method (SNM), see Algorithm 1. The computation of both the stochastic gradient and the stochastic Hessian are done on lines 9 and 10. Another contribution is done while computing the direction for the

Algorithm 1 Stochastic Newton Method

Input: Starting parameter value (θ_0), data ($\mathcal{D}$), function (f), gradient (∇f), Hessian ($\nabla^2 f$), number of epochs (n_{ep}), batch size (n_{batch})
Output: Epochs (e), parameters (θ), function values (f_v)

function SNM
  $(n_{\mathcal{D}}, m) = |\mathcal{D}|$ ▷ Number of samples and parameters
  $n_{iter} \leftarrow \lceil n_{ep} n_{\mathcal{D}} / n_{batch} \rceil$ ▷ Number of iterations
  Initialize e, θ and f_v. Set $\theta[0] \leftarrow \theta_0$
  for $i = 0 \dots n_{iter}$ **do**
    $e[i] \leftarrow i \cdot n_{batch} / n_{\mathcal{D}}$ ▷ Store the epoch
    $f_v[i] \leftarrow f(\theta[i])$ ▷ Store the function value
    `idx` $\leftarrow n_{batch}$ values from $\mathcal{U}(0, n_{\mathcal{D}})$ without replacement
    `grad` $\leftarrow \nabla f_{\texttt{idx}}(\theta[i])$ ▷ Gradient on the samples from `idx`
    `hess` $\leftarrow \nabla^2 f_{\texttt{idx}}(\theta[i])$ ▷ Hessian on the samples from `idx`
    if `hess` is non singular **then**
      `inv_hess` $\leftarrow$ `hess`$^{-1}$
      `step` $\leftarrow -$`grad`$\cdot$`inv_hess`
    else
      `step` $\leftarrow$ `grad`
    $\alpha \leftarrow$ Backtracking Line Search with `step` on the subset of data with indices from `idx`
    $\theta[i+1] \leftarrow \theta[i] + \alpha \cdot$ `step`
  $e[n_{iter}] \leftarrow n_{iter} \cdot n_{batch} / n_{\mathcal{D}}$
  $f_v[n_{iter}] \leftarrow f(\theta[n_{iter}])$
  return e, θ and f_v

next step. Indeed, with small batches, the Hessian may be singular. For example, it is possible that a parameter was not present with a chosen alternative. Therefore, the row and column of the Hessian will both be zero for this particular parameter, thus making it singular. The countermeasure to this possibility is to test if the Hessian is singular or not. If it is not the case, then the algorithm performs a standard Newton step with the stochastic Hessian and gradient. However, if the Hessian is singular, the algorithm performs a Stochastic Gradient Descent (SGD) step. Concerning the choice of the learning rate, for a given objective function, it often differs between SGD and Newton Method. Therefore, the algorithm should either receive two different learning rate or we can perform a line search, as explained at the end of Section III-B.

IV. Results

☞ **NOTE:**
- First order methods on normalized vs unormalized data
- First order methods depending on batch size
- Quasi Newton methods depending on batch size
- Second order methods depending on batch size
- Percentage of grad vs newton step

V. Discussion

VI. Conclusion

VII. Acknowledgements

☞ **NOTE:** Thanks Tim!

References

[1] M. Bierlaire, K. Axhausen, and G. Abay, "The acceptance of modal innovation: The case of Swissmetro," *Swiss Transport Research Conference 2001*, Mar. 2001. [Online]. Available: https://infoscience.epfl.ch/record/117140

[2] M. Bierlaire, "BIOGEME: a free package for the estimation of discrete choice models," *Swiss Transport Research Conference 2003*, Mar. 2003. [Online]. Available: https://infoscience.epfl.ch/record/117133

[3] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv:1609.04747 [cs]*, Sep. 2016, arXiv: 1609.04747. [Online]. Available: http://arxiv.org/abs/1609.04747

[4] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011. [Online]. Available: http://jmlr.org/papers/v12/duchi11a.html

[5] R. Fletcher, *Practical Methods of Optimization; (2Nd Ed.)*. New York, NY, USA: Wiley-Interscience, 1987.

[6] A. Mokhtari and A. Ribeiro, "RES: Regularized Stochastic BFGS Algorithm," *IEEE Transactions on Signal Processing*, vol. 62, no. 23, pp. 6089–6104, Dec. 2014.

[7] J. Caswell, "A treatise of algebra, both historical and practical : with some additional treatises I. of the cono-cuneus; being a body representing in part a conus, an part a cuneus ; II. of angular sections; and other things relating there unto, and to Trigonometry ; III. of the angle of contact; with other things appertaining to the composition of magnitudes, the inceptive of magnitudes, and the composition of motions, with the results thereof ; IV. of combination, alternations, and aliquot parts," Tech. Rep., 1685.

[8] L. Armijo, "Minimization of functions having Lipschitz continuous first partial derivatives," *Pacific Journal of Mathematics*, vol. 16, no. 1, pp. 1–3, Jan. 1966. [Online]. Available: https://msp.org/pjm/1966/16-1/p01.xhtml