CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

On Optimization Algorithms for Maximum Likelihood Estimation

Anh Tien Mai
Fabian Bastin
Michel Toulouse

December 2014

CIRRELT-2014-64

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

On Optimization Algorithms for Maximum Likelihood Estimation

Anh Tien Mai[1,*], Fabian Bastin[1], Michel Toulouse[1,2]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

[2] Vietnamese-German University, Le Lai Street, Hoa Phu Ward, Binh Duong New City, Binh Duong Province, Vietnam

Abstract. Maximum likelihood estimation (MLE) is one of the most popular technique in econometric and other statistical applications due to its strong theoretical appeal, but can lead to numerical issues when the underlying optimization problem is solved. We examine in this paper a range of trust region and line search algorithms and focus on the impact that the approximation of the Hessian matrix has on their respective performance. In particular, we propose new methods to switch between the approximation schemes and compare the effectiveness of these strategies with existing approaches. We assess the numerical efficiency of the proposed switching methods for the estimation of discrete choice models, more precisely mixed logit and logit based route choice models.

Keywords: Trust region, line search, maximum likelihood estimation, Hessian approximation, switching algorithms, discrete choice, mixed logit, route choice.

Acknowledgements. This research was partially funded by Natural Sciences and Engineering Research Council of Canada (NSERC).

---

* Corresponding author: AnhTien.Mai@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2014

1 Introduction

Trust region and line search techniques, originally introduced as a globalization of the locally-converging Newton technique, are currently among the most popular techniques for optimizing possibly non-convex, twice-continuously differentiable non-linear functions. In this setting, the methods typically rely on a second-order Taylor-development of the objective function, therefore requiring the Hessian of the objective function to be available. However, the numerical cost associated to Hessian evaluation is usually computationally expensive, and one prefers to construct some approximation of the Hessian, leading to so-called quasi-Newton techniques. The most popular Hessian approximations are BFGS (Broyden, 1970, Fletcher, 1970, Goldfarb, 1970, Shanno, 1970) and symmetric rank-1 (SR1) update (Conn et al., 1991), both of them maintaining symmetry of the matrix and satisfying the secant condition. The convergence to the true Hessian can however be slow, resulting in numerous iterations during the optimization process.

This work focuses on maximum likelihood estimation problems, aiming to investigate efficient optimization algorithms to solve them. An alternative Hessian approximation has been proposed in this context by Berndt et al. (1974). This approximation, called BHHH by reference to the authors, relies on the information identity property, and appears to be less computationally demanding, while it better reflects the problem structure. This explains the popularity of the approach, as illustrated for instance in Train (2009, Chapter 8). Unfortunately, the conditions needed to ensure validity of the information identity are difficult to satisfy, especially as they require a correctly specified model. In practice, these conditions are often violated, and the estimation can fail to converge. This has led Bunch (1987) to consider the log-likelihood problem as a particular case of generalized regression and to propose to add a correction term to the Hessian approximation, similarly to the Gauss-Newton method in the context of least-squares problems (Dennis Jr et al., 1981).

In Bunch's original proposal, a single switch was to be executed from a first quadratic model to a second one once the first one no longer converge effectively. An issue then was to identify the iteration where the switch had to be performed. In the approach we propose, the switch is considered at each iteration of the optimization method, raising the question of how to select among a set of Hessian approximations at each iteration. The present paper addresses this issue by proposing new criteria for switching between quadratic models, either to build a subproblem in trust region methods or to compute the search-direction in line search methods. More specifically, we propose two new models that differ in the way the Hessian approximation is selected at each iteration. The *predictive* model proposes a way to predict a Hessian

approximation for the next iteration by considering the accurateness of the quadratic models. This model applies to both trust region and line search methods. The *multi-subproblems* model is designed for trust region methods only, in which several subproblems are taken into account and solved approximately. This model selects the Hessian approximation that decreases the most the objective function. The proposed optimization algorithms are applied to mixed logit models and logit based route choice models.

The paper is structured as follows. We first provide in Section 2 relevant background on maximum likelihood estimation. Section 3 briefly describes the trust region and line search optimization methods, and Section 4 introduces different Hessian approximation methods. We present our switching strategies for these two optimization methods in Section 5. Section 6 introduces some basic concepts of discrete choice theory, as it constitutes the studied framework for our numerical experiments. Numerical assessments are reported in Section 7 and finally Section 8 concludes.

2 Maximum likelihood estimation

Maximum likelihood is one the most popular technique in statistics to estimate the parameters of a model, given some observations assumed to be the realizations of some random vector. More precisely, consider a random vector Y, and assume we have N observations independently drawn from this vector. Let assume for now that Y is continuous. Denote by $f(Y|\theta)$ the probability density function (pdf) Y, conditioned on a set of parameters θ. The random distribution would be completely characterized if we knew the particular value of θ, say θ_0, corresponding to the population under interest. In the discrete case, we would consider the probability mass function instead of the density. Since the observations are assumed to be independent, the joint density is the product of the individual densities:

$$f(y_1, y_2, \ldots, y_N \,|\, \theta) = \prod_{n=1}^{N} f(y_n|\theta).$$

However, we are not interested in the observations, that are known, but rather in θ, so it is convenient to consider a function of θ that would follow the value of the joint density, given the observations $y_1, \ldots, y_N$:

$$L(\theta \,|\, y_1, y_2, \ldots, y_N) = f(y_1, y_2, \ldots, y_N \,|\, \theta),$$

where $L(\theta \,|\, y_1, y_2, \ldots, y_N)$ is called the likelihood function. Since we do not know θ_0, we will approximate it by computing an estimator $\hat{\theta}_N$ of its value, that can be judged

as the most likely value for θ, given our observations. This is done by maximizing the function $L(\theta \,|\, y_1, \ldots, y_N)$ with respect to θ:

$$\max_{\theta \in \Theta} L(\theta \,|\, y_1, y_2, \ldots, y_N), \tag{1}$$

where we confine the search to the parameter space Θ, and we assume that θ_0 belongs to Θ. We assume furthermore that (1) has a unique solution, called the maximum likelihood estimator:

$$\hat{\theta}_N = \arg\max_{\theta \in \Theta} L(\theta \,|\, y_1, y_2, \ldots, y_N).$$

In practice, due to numerical stability issues, it is often more convenient to work with the logarithm of the likelihood function, called the log-likelihood:

$$\ln L(\theta \,|\, y_1, \ldots, y_N) = \ln L(\theta \,|\, y_1, \ldots, y_N) = \sum_{n=1}^{N} \ln f(y_n|\theta) \tag{2}$$

or the average log-likelihood

$$\frac{1}{N} \sum_{n=1}^{N} \ln f(y_n|\theta). \tag{3}$$

The likelihood function can be denoted simply by $L(\theta)$, and its logarithm by $LL(\theta)$. Maximizing the log-likelihood is equivalent to maximize the likelihood since the logarithm operator is monotonically increasing:

$$\hat{\theta}_N = \arg\max_{\theta \in \Theta} LL(\theta),$$

assuming that the solution exists and is unique. The maximum likelihood estimator is attractive as, under mild conditions, $\hat{\theta}_N$ almost surely converges to θ_0 as N grows to infinity, and the distribution function of $\sqrt{N}(\hat{\theta}_N - \theta_0)$ converges to the multinormal distribution function with mean zero and variance-covariance matrix $\mathcal{V}$. The reader is referred e.g. to Newey and McFadden (1986) for more details.

3 Optimization algorithms

This section describes algorithms for solving the maximum likelihood (or log-likelihood) estimation problem. This problem can be expressed as a unconstrained non-linear optimization problem as follows:

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

where $f(x) = -L(x)$ or $-LL(x)$ is a general notation of the likelihood or log-likelihood function, and we use x instead of θ in order to follow conventional notation in optimization. We seek optimization algorithms for this problem that behave in the following manner:

1. Reliably converge to a local minimizer from an arbitrary starting point;
2. Do so as quickly as possible.

Algorithms which satisfy the first above requirement are called globally convergent.

Most optimization algorithms use the value of the objective function f and possibly its first and second derivatives. Since the evaluation of the true Hessian is usually computationally expensive, approximations of the Hessian are often preferred, with the hope of retaining fast local convergence at a lower cost. We first review two classes of optimization algorithms that satisfy the two conditions above and where the Hessian or its approximation play an important role: *line search methods* and *trust region methods*. Next, we describe several methods for approximating the Hessian matrix.

3.1 Line search methods

Line search methods are effective iterative algorithms to compute local minimizers in unconstrained optimization problems. Each iteration k of a line search algorithm computes a search direction p_k and a positive step length α_k along the search direction that satisfies a sufficient decrease in the function as measured by the inequality

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^T p_k \tag{4}$$

for some constant c_1. This condition is the first of the *Wolfe conditions*, also called *Armijo condition*. In the context of line search, this condition could be satisfied for all sufficiently small values of α, so it may not be enough by itself to ensure fast convergence, or even convergence to a local solution. Thus, another condition is proposed, called the *curvature condition*:

$$\nabla f(x_k + \alpha_k p_k)^T p_k \leq c_2 \nabla f(x_k)^T p_k \tag{5}$$

for some constant c_2 satisfying $c_1 \leq c_2 < 1$. (5) is sometimes replaced by the strong curvature condition

$$|\nabla f(x_k + \alpha_k p_k)^T p_k| \geq c_2 |\nabla f(x_k)^T p_k|,$$

yielding the *strong Wolfe condition*. Once a scalar α_k has been found satisfying the (strong) Wolfe conditions, x_{k+1} is set to $x_k + s_k$, where $s_k = \alpha_k p_k$ is the accepted step at k^{th} iterate.

Most line search algorithms require the search direction p_k to be a descent direction, i.e. $p_k^T \nabla f(x_k) < 0$, thus reducing the function $f(x)$ along this direction. In the steepest descent approach, the search direction is simply the opposite of the gradient $p_k = -\nabla f(x_k)$. Newton's method or quasi-Newton methods compute p_k by minimizing the predictive quadratic model

$$m_k(p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T H_k p, \tag{6}$$

leading to $p_k = -H_k^{-1} \nabla f(x_k)$, where H_k is a symmetric and non-singular matrix. In Newton's method, H_k is the exact Hessian $\nabla^2 f(x_k)$, but in quasi-Newton method, H_k is an approximation of the Hessian updated at every iteration of a line search algorithm. When p_k is defined in this way and the matrix H_k is positive definite, we have

$$p_k^T \nabla f(x_k) = -\nabla f(x_k)^T H_k^{-1} \nabla f(x_k) < 0$$

and therefore p_k is a descent direction.

3.2 Trust region methods

Trust region methods approach global optimization by (approximately) minimizing, at each iteration, a model of the objective function in a region centered at the current iterate, defined as

$$\mathcal{B}_k = \{s \in \mathbb{R}^n \text{ and } ||s||_k \leq \Delta_k\}.$$

Here Δ_k is a scalar known as the trust region radius and $||\cdot||_k$ is some norm, possibly iteration-dependent. An usual choice is the 2-norm. The model is typically chosen as a quadratic approximation of the objective function, such as the n-dimensional quadratic model m_k defined in (6). In other words, to get the next iterate, the step s_k is found by solving the following constrained optimization problem:

$$\min_{s \in \mathcal{B}_k} \{m_k(s)\}.$$

This is also called the subproblem of the trust region algorithm. The exact minimization of the subproblem is often expensive and unnecessary, so instead it can be solved approximately using less computational time, using for instance the Steihaug-Toint algorithm (Steihaug, 1983, Toint, 1981).

The main idea of trust region methods is then to compare the decrease predicted by the model minimization with the actual decrease of the objective function, computing the ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{f(x_k) - m_k(s_k)}.$$

If the agreement ρ_k is sufficiently good, the trial point becomes the new iterate and the trust region is maintained or enlarged. In such a case, the iteration is said to be successful or very successful, depending of the magnitude of ρ_k. If this agreement is poor, the trust region is shrunk in order to improve the quality of the model. We refer the reader to Conn et al. (2000) or Nocedal and Wright (2006, Chapter 4) for more details.

4 Hessian approximation methods

Line search and trust region methods therefore typically make extensive use of the quadratic model m_k which is strongly based on the Hessian matrix H_k. Because the computation of the exact Hessian is often too expensive, several Hessian approximation methods have been proposed. We now describe some well-known approaches.

4.1 Secant approximations

Each iteration k of the secant method uses the curvature information from the current iteration, and possibly the matrix H_k to define H_{k+1}. The matrix H_{k+1} is computed to satisfy the secant equation

$$H_{k+1} d_k = y_k$$

where $d_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. In this way, $H_{k+1} d_k$ is a finite difference approximation to the derivative of $\nabla f(x)$ in the direction of $x_{k+1} - x_k$. To determine H_{k+1} uniquely, an additional condition is imposed that among all symmetric matrices satisfying the secant equation in some sense, the one closest to the current matrix H_k is selected:

$$\min_{H = H^T, H d_k = y_k} ||H - H_k||_W$$

where $||\cdot||_W$ is the weighted Frobenius norm: $||A||_W = ||W^{\frac{1}{2}} A W^{\frac{1}{2}}||_F$ in which $||\cdot||_F$ is defined by $||C||_F = \sqrt{\sum_{1 \leq i,j \leq n} c_{ij}^2}$. The weight W can be chosen as a matrix

satisfying the condition $Wy_k = d_k$. This condition allows an easy solution of the problem above, the unique solution being

$$H_{k+1} = H_k - \frac{H_k d_k d_k^T H_k}{d_k^T H_k d_k} + \frac{y_k y_k^T}{y_k^T d_k} \qquad \text{(BFGS)}. \tag{7}$$

This update is also called the BFGS (or rank-2) update (Broyden, 1970, Fletcher, 1970, Goldfarb, 1970, Shanno, 1970), and is one of the most popular Hessian approximation method.

Another well-known approximation matrix is the symmetric rank-1 (SR1) update which maintains the symmetry of the matrix but does not guarantee positive definiteness, allowing to take advantage of a negative curvature. The SR1 update also complies with the secant equation,

$$H_{k+1} d_k = y_k,$$

with the additional requirement

$$H_{k+1} = H_k \pm \delta\delta^T.$$

The only formula that satisfies these conditions is given by

$$H_{k+1} = H_k + \frac{(y_k - H_k d_k)(y_k - H_k d_k)^T}{(y_k - H_k d_k)^T d_k} \qquad \text{(SR1)}.$$

For a detailed description see Conn et al. (1991). Since this Hessian approximation is not necessarily positive definite, the quadratic model (6) can be unbounded below. This is not an issue for trust region methods as the search space is bounded at each iteration, but can lead to failure of line search methods, requiring modifications of the algorithms (Öztoprak and Birbil, 2011).

4.2 Statistical approximation

When maximizing the log-likelihood to estimate model parameters, a specific Hessian approximation can be derived, reflecting the problem structure. If the model is correctly specified and assuming that θ_0 is the true parameters vector, we have the information matrix equality

$$I(\theta_0) = -\mathbb{E}[\nabla^2 f(y|\theta_0)],$$

where $I(\theta_0)$ is the Fisher information matrix, defined as the covariance matrix of the score at θ_0, and the expectation is taken over the population. The score is defined as

$$g(y|\theta) = \nabla_\theta \ln f(y|\theta),$$

leading to the following expression for the information matrix

$$I(\theta_0) = E[\nabla_\theta \ln f(y|\theta_0) \nabla_\theta^T \ln f(y|\theta_0)]$$

where T is the transpose operator. For a finite sample, the information matrix can be consistently estimated as

$$I_N(\theta_N^*) = \frac{1}{N} \sum_{n=1}^{N} \nabla_\theta [\ln f(y_n|\theta_N^*) \nabla_\theta \ln f(y_n|\theta_N^*)^T].$$

Berndt et al. (1974) suggest to extrapolate on the information identity, using the opposite of the (sample) information matrix as the Hessian approximation:

$$H_{BHHH}(\theta) = -I_N(\theta). \tag{8}$$

This approximation is known as the BHHH approximation or statistical approximation, and, being positive definite, can be used at each iteration of the trust region or line search algorithms. It only requires the information available at the current iteration, and is cheap to obtain. Moreover, as it relies on the specific properties of the maximum log-likelihood problem, the BHHH approximation is often closer to the true Hessian than the secant approximations, especially during the first iterations. The secant approximations only asymptotically converge, under some conditions, to the true Hessian with the number of iterations. However, two issues affect the use of the BHHH approximation. First, the information matrix equality is only valid asymptotically with the number of observations, at the true parameters. Second, it requires a correctly specified model, which can be very difficult to obtain. Therefore, the BHHH approximation may not converge to the Hessian of the log-likelihood objective function, sometimes leading to poor performances, especially when close to the solution.

4.3 Corrected BHHH approximations

A closer look at the log-likelihood Hessian exhibits more clearly the rule of the BHHH approximation, and suggests some corrective procedures to enforce convergence. Writing again the log-likelihood function

$$LL(\theta) = \frac{1}{N} \sum_{n=1}^{N} \ln f(y_n|\theta)$$

we can derive the Hessian as

$$\nabla^2 LL(\theta) = -\frac{1}{N}\sum_{n=1}^{N}\frac{\nabla f(y_n|\theta)\nabla f(y_n|\theta)^T}{f^2(y_n|\theta)} + \frac{1}{N}\sum_{n=1}^{N}\frac{\nabla^2 f(y_n|\theta)}{f(y_n|\theta)}. \tag{9}$$

Using (8), (9) can be rewritten as

$$\nabla^2 LL(\theta) = H_{BHHH}(\theta) + A(\theta),$$

with

$$A(\theta) = \frac{1}{N}\sum_{n=1}^{N}\frac{\nabla^2 f(y_n|\theta)}{f(y_n|\theta)}.$$

The computation of $A(\theta)$ requires the calculation of N individual Hessian matrices, which is often very expensive. $A(\theta)$ however can be approximated by investigating its structure, as done in Bunch (1987). More precisely, assuming that at iteration k the matrix H_k is available to approximate the next Hessian H_{k+1}, the new approximation can be obtained by specifying an appropriate secant condition, which takes the form

$$H_{k+1}d_k = y_k, \tag{10}$$

in which H_{k+1} is a new matrix approximation. We can write

$$H_{k+1} = H_{BHHH}(\theta_{k+1}) + A_{k+1},$$

where A_{k+1} is an approximation of $A(\theta_{k+1})$.

Bunch (1987) proposes two secant equations to approximate $A(\theta)$. First, (10) gives $(H_{BHHH}(\theta_{k+1}) + A_{k+1})d_k = y_k$, and by setting $\bar{y}_k^1 = y_k - H_{BHHH}(\theta_{k+1})d_k$, this yields the secant equation

$$A_{k+1}d_k = \bar{y}_k^1 \tag{11}$$

which can be used to approximate matrix A_{k+1}. The second secant equation is derived by approximating each individual Hessian matrix $\nabla^2 f(y_n|\theta)$. More precisely, we note that

$$\nabla^2 f(y_n|\theta_k)d_k \approx \nabla f(y_n|\theta_{k+1}) - \nabla f(y_n|\theta_k).$$

Substitution into (9) gives

$$A(\theta_k)d_k \approx \frac{1}{N}\sum_{n=1}^{N}\frac{\nabla f(y_n|\theta_{k+1}) - \nabla f(y_n|\theta_k)}{f(y_n|\theta_k)}$$

So if we define $\bar{y}_k^2 = \frac{1}{N}\sum_{n=1}^{N} \frac{\nabla f(y_n|\theta_{k+1}) - \nabla f(y_n|\theta_k)}{f(y_n|\theta_k)}$, the second secant approximation can be written as

$$A_{k+1} d_k = \bar{y}_k^2. \tag{12}$$

Bunch (1987) suggests to update A_{k+1} with the BFGS method, but any secant approximation can be used, for instance the SR1 update.

5 Model switching strategies

The objective of this work is to obtain computationally efficient optimization methods for solving the maximum likelihood estimation problem. We propose approaches based on the line search and the trust region methods as these two methods methods are globally convergent, using a quadratic model of the log-likelihood function. As previously discussed, several Hessian approximation methods are available, with performances that may vary in different phases of the optimization process. Some authors have considered switching among Hessian approximations during the optimization. For example, Phua and Setiono (1992) proposed a switching algorithm based on the condition number of the secant approximation matrices. Bunch (1987) proposed an approach called *model switching* which initially uses the BHHH approximation and then switches to a corrected BHHH approximation in the last phase of the optimization process. This section introduces new and switching strategies adapted to line search and trust region methods. At first we present, in the following, a general framework for the switching models so that our algorithms can be described explicitly.

We denote by $\mathcal{H}_k$ the set of available Hessian approximations to select from at the k^{th} iteration of an optimization algorithm:

$$\mathcal{H}_k = \{H_k^i \, , i = 1, \ldots\},$$

where H_k^i refers to a specific Hessian approximation. For example the matrix obtained by the BHHH approximation can be denoted by H_k^1, the matrix obtained by the BFGS method can be denoted by H_k^2, etc. Each iteration of an optimization algorithm with model switching executes one more step in which one Hessian approximation is chosen from $\mathcal{H}_k$ in order to compute the search direction in a line search algorithm or to define the subproblem in a trust region algorithm. The next two sections describe our switching strategies.

5.1 Multi-subproblems model

Each iteration of a trust region algorithm defines a subproblem $\min_{s\in\mathcal{B}_k}\{m_k(s)\}$ where $\mathcal{B}_k$ and $m_k(s)$ are respectively the trust region vectors and a quadratic model used to approximate the objective function at iteration k. Solving approximately this subproblem determines the current step. Given a set of Hessian approximations $\mathcal{H}_k$ at each iteration, there is a set of corresponding subproblems:

$$\min_{s\in\mathcal{B}_k} m_k^i(s) = \min_{s\in\mathcal{B}_k}\left\{f(x_k) + \nabla f(x_k)^T s + \frac{1}{2}s^T H_k^i s\right\}, \quad H_k^i \in \mathcal{H}_k. \tag{13}$$

We solve approximately all the available subproblems in order to obtain the set of steps $\{s_k^i\}$ and to choose a step $s_k^{i^*}$ which satisfies

$$i^* \in \arg\min_i f(x_k + s_k^i). \tag{14}$$

This approach evaluates the decrease in the objective function made by each proposed step and selects the subproblem which maximizes this decrease. A trust region method with the multi-subproblems switching strategy is described in Algorithm 1. We note that Algorithm 1 requires solving all the subproblems, therefore calculating more than one objective function value at each iteration. In the numerical experiments section, we explicit the stopping criteria used.

5.2 Predictive model

Consider a set of models as in (13) and denote by $\delta_k^i(s)$ the absolute difference between the quadratic model $m_k^i(s)$ and $f(x_k + s)$. We call $\delta_k^i(s)$ the approximation error of the quadratic model $m_k^i(s)$. The predictive model uses this quantity to evaluate the accurateness of the quadratic model and to select a Hessian approximation for the next iteration. More precisely, at the end of an iteration k with step s_k of some optimization algorithm (either trust region of line search), and given that the objective function $f(x_k + s_k)$ is already computed, the approximation errors associated with different Hessian approximations can be computed as

$$\delta_k^i(s_k) = |f(x_k + s_k) - m_k^i(s_k)| = \left|f(x_k + s_k) - f(x_k) - s_k^T\nabla f(x_k) - \frac{1}{2}s_k^T H_k^i s_k\right|.$$

Consequently, the next Hessian approximation $H_{k+1}^{i^*}$ is predicted by minimizing this error

$$i^* = \arg\min_i\{\delta_k^i(s_k)\}. \tag{15}$$

Algorithm 1 Trust region method with the multi-subproblems switching model

Step 0. Initialization: Given an initial point x_0, an initial trust region with radius Δ_0 and constants η_1, η_2, γ which satisfy

$$1 > \eta_1 > \eta_2 > 0 \text{ and } 1 > \gamma_1 > \gamma_2 > 0,$$

choose an initial matrix H_0 and set $k = 0$.

Step 1. If stopping criteria are met, stop. Otherwise, go to Step 2.

Step 2. Define a set of Hessian approximations $\mathcal{H}_k$.

Step 3. Step calculation: Calculate the set of steps $\{s_k^i,\, i = 1, 2 \ldots\}$ by solving approximately all the subproblems.

$$\min_{s \in \mathcal{B}_k} \{m^i(s),\ H_k^i \in \mathcal{H}_k\}.$$

Determine the best step $s_k^{i^*}$ by solving (14).
Compute the ratio ρ_k

$$\rho_k = \frac{f(x_k) - f(x_k + s_k^{i^*})}{f(x_k) - m_k(s_k^{i^*})}.$$

If $\rho_k > \eta_2$ set $x_{k+1} = x_k + s_k$, otherwise set $x_{k+1} = x_k$.

Step 4. Trust region radius update: Update the trust region radius as follows:

$$\Delta_{k+1} = \begin{cases} \max\{2\|s_k\|, \Delta_k\} & \text{If } \rho_k \geq \eta_1 \\ \gamma_1 \Delta_k & \text{If } \eta_1 > \rho_k \geq \eta_2 \\ \gamma_2 \Delta_k & \text{If } \rho_k \leq \eta_2 \end{cases}$$

Set $k \leftarrow k + 1$ and go to Step 1.

The predictive switching strategy has the advantage of not requiring any new evaluation of the objective function. The objective function for probabilistic choice models is often costly to evaluate, particularly with large real data sets. Avoiding the evaluation of this function improves the computational efficiency of the optimization methods. A trust region algorithm with the predictive switching strategy is described in Algorithm 2. A line search algorithm with the predictive switching strategy is described in Algorithm 3.

Algorithm 2 Trust region method with the predictive switching model

Steps 0–2. Identical to **Algorithm 1**.

Step 3. Step calculation: Evaluate the step s_k by solving approximately the subproblem

$$\min_{x_k+s\in\mathcal{B}_k} m_k(s).$$

Evaluate ρ_k

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{f(x_k) - m_k(s_k)}.$$

If $\rho_k > \eta_2$ set $x_{k+1} = x_k + s_k$, otherwise set $x_{k+1} = x_k$.

Step 3. Hessian approximation prediction: The next Hessian approximation $H_{k+1}^{i^*}$ is predicted by solving (15). Set $H_{k+1} = H_{k+1}^{i^*}$.

Step 4. Trust region radius update: Identical to Step 4 of Algorithm 1. Set $k \leftarrow k + 1$ and go to Step 1.

6 Discrete choice theory

The proposed switching techniques have been applied on the estimation of various discrete choice models, so before describing our numerical experiments, we provide a short introduction to discrete choice theory.

6.1 Discrete choice models

Discrete choice theory examines how people make decisions among a finite number of possible choices. More specifically, we consider a set of N individuals, each one having

Algorithm 3 Line search method with the predictive switching model

Step 0. Initialization: Choose an initial matrix H_0 and set $k = 0$.

Step 1. If stopping criteria are met, stop. Otherwise, go to Step 2.

Step 2. Search direction calculation: Compute search direction p_k which satisfies the equation:

$$H_k p_k = -\nabla f(x_k).$$

Step 3. Step calculation: Compute step length α_k which satisfies the *Wolfe conditions* and set $x_{k+1} = x_k + \alpha_k p_k$.

Step 4. Hessian approximation prediction: Predict the next Hessian approximation $H_{k+1}^{i^*}$ using (15). Set $H_{k+1} = H_{k+1}^{i^*}$.

Set $k \leftarrow k + 1$ and go to Step 1.

to choose one alternative within a finite set. The Random Utility Maximization (RUM) theory (McFadden, 1973) assumes that each individual n associates to each alternative i within a choice set C_n an utility U_{ni}. This utility consists of two parts: a deterministic part V_{ni} known by the modeler and an uncertain part ϵ_{ni} which is known to individuals but unknown to modelers. The instantaneous utility is

$$U_{ni} = V_{ni} + \epsilon_{ni}.$$

The deterministic term V_{ni} can include attributes of the alternative as well as socio-economic characteristics of the individual. In general a linear-in-parameters is used: $V_{ni} = \beta^T x_{ni}$ where β is a vector of parameters to be estimated and x_{ni} is the vector of attributes of alternative i as observed by individual n. The decision maker aims to maximize the instantaneous utility so the probability that an alternative i is chosen by individual n is

$$P(i|n, C_n) = P(U_{ni} \geq U_{nj},\ \forall j \in C_n) = P(V_{ni} + \epsilon_{ni} \geq V_{nj} + \epsilon_{nj},\ \forall j \in C_n)$$

Different assumptions for the random terms $\epsilon_{nj},\ j \in C_n$ can lead to different types of discrete choice models. A popular model is the multinomial logit (MNL) model which assumes that the random terms are independent and identically distributed (i.i.d.) Extreme Value type I with mean μ and scale factor λ (often normalized to

one), characterized by the distribution function

$$F(x) = e^{-e^{-\lambda(x-\mu)}}.$$

The choice probability is then

$$P_L(i|n, C_n) = \frac{e^{\lambda V_{ni}}}{\sum_{j \in C_n} e^{\lambda V_{nj}}}. \tag{16}$$

Such model can be estimated by maximizing the log-likelihood function over the parameters vector β:

$$\max_{\beta} LL(\beta) = \frac{1}{N} \sum_{n=1}^{N} \ln P_L(i|n, C_n).$$

The MNL model provides a simple closed form for the choice probabilities. It however has an important limitation which is the independence of irrelevant alternatives (IIA) property (see for instance Train, 2009). Other models have been proposed in order to relax this limitation. Examples are the nested logit model (Ben-Akiva and Lerman, 1985), the mixed logit model (McFadden, 1978) and the cross-nested logit models (McFadden, 1981, Vovsha and Bekhor, 1998). In the following we briefly describe the mixed logit and logit route choice models, which are used in our numerical tests.

6.2 Mixed logit models

Mixed logit models have been known for many years but have become popular with McFadden and Train (2000). They obviate the three limitations of standard logit by allowing for random taste variation, unrestricted substitution patterns, and correlation in unobserved factors over time (Train, 2009). Mixed logit models can be derived under a variety of different behavioural specifications, where each derivation provides a particular interpretation. The first application of Mixed logit was apparently the demand for electricity-using goods (Electric Power Research Institute (1977)).

Using the random-terms formulation, we here assume that the vector of model parameters β is itself derived from a random vector ω and a parameter vector θ, which we express as $\beta = \beta(\omega, \theta)$. ω typically specifies the random nature of the model and the vector parameters θ quantifies the population characteristic for the

model. The associated unconditional probability is obtained by integrating (16) over ω:

$$P_{ML}(i|n, C_n, \theta) = E_P[P_L(i|n, C_n, \omega, \theta)] = \int P_L(i|n, C_n, \omega, \theta) f(\omega) d\omega \qquad (17)$$

where P is the probability measure associated to ω, E is the expectation operator and f is the density function. When T_n observations are available par individual, the correlation is often captured by assuming that the parameters β do not vary for the same individual, while being randomly distributed throughout the population (Revelt and Train, 1998). (17) then becomes

$$P_{ML}(i|n, C_n, \theta) = E_P \left[\prod_{t=1}^{T_n} P_L(i_t|n, C_n, \omega, \theta) \right], \qquad (18)$$

where i_t is the t^{th} observed choice.

It is usual to replace the expectation by some approximation, typically obtained by sampling over ω. (18) becomes

$$P_{ML}(i|n, C_n, \theta) \approx SP^R_{ML}(i|n, C_n, \theta) = \frac{1}{R} \sum_{r_i=1}^{R^n} \prod_{t=1}^{T_n} P_L(i_t|n, C_n, \omega_{r_i}, \theta)$$

where R^n is the number of random draws associated with individual n. The sample can be generated by standard Monte Carlo or quasi-Monte Carlo techniques, though there is no clear advantage of one approach in this context (Munger et al., 2012).

6.3 Logit based route choice models

Discrete choice models are also used for analyzing and predicting route choices in various transport applications. The route choice problem in real networks is characterized by a very large number of path alternatives and in practice it is not possible to enumerate all paths connecting a given origin-destination pair in a real network. In order to consistently estimate a route choice model, either paths have to be sampled (Frejinger et al., 2009), or the recursive logit (RL) model recently proposed by Fosgerau et al. (2013) can be used. These two modeling approaches have in common that they are based on the MNL model, but we will restrict ourselves to path-based models in our numerical tests.

When the path choice sets are sampled, for each individual n and a sampled choice set D_n, the probability that a path σ is chosen is

$$P(\sigma|n, D_n) = \frac{e^{V_{n\sigma} + \ln \pi(D_n|\sigma)}}{\sum_{j \in D_n} e^{V_{nj} + \ln \pi(D_n|j)}},$$

where V_{nj} is the deterministic utility of path j observed by the individual n and $\ln \pi(D_n|j)$ is the correction for sampling bias. $\pi(D_n|j)$ is the probability of sampling choice set D_n given that j is the chosen alternative (Frejinger et al., 2009). Mai et al. (2014) show that when the models are correctly specified, the information matrix equality holds if and only if the sampling corrections $\pi(D_n|j)$, $j \in D_n$, are added to the choice probabilities.

In order to deal with the overlapping of paths in the network, the path size attribute has been proposed in Ben-Akiva and Bierlaire (1999) as an additional deterministic attribute for the utilities. Frejinger et al. (2009) propose a heuristic sampling correction of the path size attribute called expanded path size (EPS). With the EPS attribute, the path choice probability is

$$P(\sigma|D_n) = \frac{e^{V_{n\sigma}+\beta_{PS}EPS_n(\sigma)+\ln \pi(D_n|\sigma)}}{\sum_{j\in D_n} e^{V_{nj}+\beta_{PS}EPS_n(\sigma)+\ln \pi(D_n|j)}}$$

where $EPS_n(\sigma)$ is the EPS attribute of path σ observed by individual n. The EPS attribute can be computed based on the length of links lying of the corresponding path and the expanded factors (Frejinger et al., 2009).

We note that the model is based on the MNL model in spite of the fact that error terms are believed to be correlated due to the physical overlap among paths, the EPS attribute allows to partly address this issue. If the random terms are correlated, the model is misspecified, leading to potential issues when relying on the BHHH technique. Modeling issues can also be present in the deterministic part of the utilities. The reader can consult Mai et al. (2014) for a discussion about the invalidity of the information matrix equality for logit based route choice models.

7 Numerical assessment

7.1 Data sets

In order to evaluate the performance of the various optimization algorithms we estimate models on three real data sets, two used with mixed models (SP2 and IRIS) and one feeding two route choice models (PS and PSL). Table 1 gives the number of observations, along with the number of individuals (in parentheses), the number of alternatives, and the number of parameters to be estimated for the four considered models. We now describe these data sets briefly.

Data set	SP2	IRIS	PS	PSL
Number of observations	2466 (2740)	2602 (871)	1832 (1832)	1832 (1832)
Number of alternatives	2	8	50	50
Number of variables	9	19	4	5

Table 1: Models

7.1.1 Mixed logit

The discrete choice data tests have been conducted on two real data sets: Cybercar (Cirillo and Xu, 2010) and IRIS (Bastin et al., 2010). Cybercar is a data set that has been collected in April 2008 at the Baltimore/Washington International Airport and concerns the operation of an automated vehicle technology called Cybercars. Our tests utilize only part of this data set which we refer to as SP2. IRIS refers to a regional transport model in Belgium where data have been collected on the propensity to switch from car to public transportation. We use this data set to evaluate the performance of the switching algorithms on a large-scale model, where statistical approximation do not work well. Seven of the explanatory variables in the IRIS model are randomly distributed, with two of them assumed to be normal or log-normal (congested and free flow time coefficients) and the remaining five are assumed to be normal. When the congested and free flow time coefficients have normal distribution we identify this model as the IN model, when they have a log-normal distribution the model is identified as the ILN model.

7.1.2 Route choice

The route choice data tests have been collected on the Borlänge network in Sweden which is composed of 3077 nodes and 7459 links. The path sample consists of 1832 trips corresponding to simple paths with a minimum of five links. There are 466 destinations, 1420 different origin-destination (OD) pairs and more than 37,000 link choices in the sample. The route choice data were collected by GPS monitoring, therefore the socio-economic information about the drivers is not available. We note that the same data have been used in other route choice modeling articles (Fosgerau et al., 2013, Frejinger, 2007). We use two path-based models with and without the EPS attribute (denoted by PL and PSL respectively). For each observation we sample a choice set of 50 draws. See Frejinger et al. (2009) and Mai et al. (2014) for details on the model specifications and estimation results.

7.2 Performance comparisons

We compare the performance of switching approaches with trust region and line search algorithms using a single Hessian approximation. For maximum likelihood estimation, the cost of model minimization in trust region methods or search direction computation in line search algorithms is typically negligible compared to the evaluation cost of the log-likelihood. Indeed, the problem dimension is usually small while the number of observations is large, as shown in Table 1. Therefore, the number of objective function evaluations captures most of the computational cost, and will be used to compare performance among algorithms. As the purpose of the comparisons is to evaluate the impact of different Hessian approximations on the performance of the optimization algorithms, the estimation results as well as the analysis on the effects of the Monte-Carlo methods will not be reported. Note that 1000 Monte Carlo random draws per individual is used for the mixed logit models. We use the Monte Carlo method to sample choice sets for the estimation of the route choice models (see for instance Mai et al., 2014). All the reported numerical results are based on 10 independent simulations. The numerical evaluations for the mixed logit models have been carried out using the package AMLET (Bastin et al., 2006). The optimization algorithms to estimate the route choice models have been implemented in MATLAB.

When a single Hessian approximation is used, trust region algorithms are implemented either with the BHHH, the BFGS or the SR1 Hessian approximation, and line search algorithms are implemented either with the BHHH or the BFGS. Line search algorithms have not been implemented with SR1 as it does not guarantee descent search directions. For the switching models, we have implemented trust region algorithms with the multi-subproblems and the predictive models, and a line search algorithm with the predictive model. In order to facilitate our discussions, we denote by BHHH^{corr1}-BFGS and BHHH^{corr2}-BFGS the corrected BHHH approximations using respectively (11) and (12), and where A_{k+1} is updated by the BFGS method. We denote by BHHH^{corr1}-SR1 and BHHH^{corr2}-SR1 the SR1 approximations based respectively on (11) and (12).

We also compare algorithms with two Bunch's switching approaches, which use a single switch from the BHHH to the BHHH^{corr1}-BFGS or to BHHH^{corr2}-BFGS. In Bunch's model switching, the algorithms start with the BHHH and do not switch to another approximation if $A(\theta_k)$ is small, meaning that a good starting guess is provided. It is however not the case for our data sets. The algorithms may take few iterations to build up A_k and switch to a corrected approximation for the remainder of the optimization procedure. Bunch however does not explain explicitly when the switch is performed. In our implementation of Bunch's approaches, we perform the switch when close to the solution. More precisely, the switch to an

alternative approximation occurs when the norm of the gradient is less than 10^{-3}.

We use the strong Wolfe conditions to compute the search directions for the line search algorithm while the sub-problems are solved by the Steihaug-Toint algorithm. For the line search algorithms, the chosen parameters for the strong Wolfe conditions are $c_1 = 10^{-4}$ and $c_2 = 0.9$. It is also a typical choice in practice. For the trust region algorithms, different parameters are chosen between the mixed logit and route choice models in order to obtain better performance (i.e. lower number of iterations) in each context. Parameters $\eta_1 = 0.9$, $\eta_2 = 0.01$ are chosen for the mixed logit models and $\eta_1 = 0.75$, $\eta_2 = 0.05$ for route choice models. We also assign $\gamma_1 = 0.7$, $\gamma_2 = 0.5$ for all the models.

Each iteration of our switching models allows to switch between several Hessian approximations, hence, the specification of a set of Hessian approximations for each model. For the multi-subproblems models, as the number of function evaluations depends on the number of matrices in the set, we only selected two approximations: BHHH and BHHH^{corr1}-BFGS. For the trust region combined with the predictive model, we selected three Hessian approximations: BHHH, BHHH^{corr1}-BFGS and BHHH^{corr1}-SR1. BHHH^{corr2}-SR1 has been selected for the predictive model in order to take advantage of a negative curvature. BHHH and BHHH^{corr1}-BFGS approximations have been selected for the line search algorithm with the predictive model, SR1 was not selected since it does not always produce a descent direction. We note that the performance of the BHHH^{corr1}- and BHHH^{corr2}- are very similar, therefore they are not included in a same set of Hessian approximations.

Considering the two switching models, the two optimization algorithms and several Hessian approximations, we have the following optimization algorithms:

[1] **TR-BHHH:** Trust region algorithm with BHHH

[2] **TR-BFGS:** Trust region algorithm with BFGS

[3] **TR-SR1:** Trust region algorithm with SR1

[4] **TR-BUNCH^1:** Bunch's switching approach with BHHH^{corr1}-BFGS

[5] **TR-BUNCH^2:** Bunch's switching approach with BHHH^{corr2}-BFGS

[6] **TR-PRED:** Trust region algorithm with the predictive model

[7] **TR-MULTI:** Trust region algorithm with the multi-subproblems model

[8] **LS-BHHH:** Line search algorithm with the BHHH

[9] **LS-BFGS:** Line search algorithm with the BFGS

[10] **LS-PRED:** Line search algorithm with the predictive model

We stop any algorithm if one of the conditions described in Table 2 is satisfied, declaring a success or a failure depending on the case encountered. Parameters MAX-ITER = 300 and $\epsilon = 10^{-5}$ were chosen for all the algorithms.

Criteria	Stopping test	Description
$\nabla f(x_k) \leq \epsilon$	GRADIENT	Successful
$\bar{\nabla} f(x_k) \stackrel{\text{def}}{=} \max_c \left(\frac{\lvert[\nabla f(x_k)]_c\rvert, \max\{[x_k]_c, 1.0\}}{\max\{\lvert f(x_k)\rvert, 1.0\}} \right) \leq \epsilon$	RELATIVE GRADIENT	Successful
$k \geq$ MAX-ITER	TOO MANY ITERATIONS	Fail
$0 < x_{k+1} - x_k \leq \epsilon$	STEP SIZE	Fail
$\Delta_k \leq \epsilon$	TRUST REGION RADIUS	Fail

Table 2: Summary of stopping conditions

Table 3 reports average numbers of function evaluations for all the algorithms. For the sake of comparison we report the average number of iterations in parentheses and the number of failures in brackets. The quantities are reported based only on successful runs and the number in bold in each column is the best result. Among trust region algorithms with a single Hessian approximation, results show that for the SP2 and IN models, BHHH approximation compared better than the secant approximations (i.e. BFGS and SR1), which explains why BHHH is often the favorite approach for MLE. For these models, the algorithms with the BHHH method always reach the optimal solution rapidly. Alternatively, the secant approximations (BFGS and SR1) perform much better than the BHHH for the route choice models when used in the trust region algorithm. This can partly be explained by the violation of the information matrix equality, which has been shown in Mai et al. (2014). For the most complex model ILN, the TR-BHHH algorithm has failed to converge for 9 of the 10 runs. In this case, the algorithm rapidly converges to the neighborhood of the solution, but then progresses very slowly close to the optimum, finally it fails to satisfy one of the successful stopping conditions. The norms of the gradient and relative gradient are always greater than 10^{-4} since the threshold for our algorithms is $\epsilon = 10^{-5}$. We however note that for the successful case the TR-BHHH performs the best compared to the other algorithms. These results translate a well-known behavior of the BHHH approximation as it may not converge to the true Hessian due to misspecification issues. On the contrary, there are no failures for the line search algorithms, even for the ILN model. Line search algorithms present the same

trends than trust region methods between BFGS and BHHH, but are sometimes faster, sometimes slower.

Algorithms		SP2	IN	ILN	PS	PSL
Trust region	TR-BHHH	27.0 (27.0)	23.9 (23.9)	37.0* (37.0) [9]	40.5 (40.5)	58.2 (58.2)
	TR-BFGS	52.9 (52.9)	155.1 (155.1)	147.6 (147.6)	19.6 (19.6)	22.5 (22.5)
	TR-SR1	42.1 (42.1)	241.5 (241.5)	238.4*(238.4) [2]	24.5 (24.5)	25.4 (25.4)
	TR-BUNCH[1]	20.6 (20.6)	33.9 (33.9)	57.4 (57.4)	51.3 (51.3)	51.0 (51.0)
	TR-BUNCH[2]	20.9 (20.9)	34.5 (34.5)	57.6 (57.6)	51.3 (51.3)	51.0 (51.0)
	TR-PRED	**14.2** (14.2)	21.8 (21.8)	**54.7** (54.7)	20.6 (20.6)	19.6 (19.6)
	TR-MULTI	46.4 (23.2)	40.4 (20.2)	77.4 (38.4)	33.2 (16.6)	31.4 (15.7)
Line search	LS-BHHH	28.1 (14.6)	**20.1** (17.6)	78.8 (46.2)	22.6 (22.1)	22.2 (21.7)
	LS-BFGS	31.8(15.8)	126.0(98.9)	202.5 (142.0)	**19.0** (17.3)	**19.1** (17.6)
	LS-PRED	34.7 (15.1)	20.5 (18.1)	70.5 (43.8)	22.6 (22.1)	22.2 (21.7)

Table 3: Performance comparison

Among the switching algorithms, for the mixed logit models, the two Bunch's approaches perform similarly and they are generally better than the trust region algorithms with a single Hessian approximation. For the route choice models they are however slower compared to other algorithms. Our predictive model with line search and trust region algorithms is slightly better than Bunch's switching algorithms as well as the classical algorithms. The results show that the predictive algorithms are always competitive, both for the trust region and the line search version, while the TR-MULTI is the slowest. This is expected as the TR-MULTI algorithm requires two evaluations of the objective function at each iteration, leading to double the number of function evaluations, while the other trust region strategies only compute the objective function once per iteration. In other words, the TR-MULTI method is usually more effective in terms of iterations, while the total cost, involving the cost per iteration, is higher.

In all previous experiments, a standard starting point ($x_0 = 0$ for the mixed logit models and $x_0 = (-3, \ldots, -3)$ for the route choice models) was chosen as the initial point of the iterative process. To evaluate the performance of the algorithms in difficult cases, we perform additional experiments on the ILN problem, the most complex model, with a starting point chosen far away from the optimum. Table 4 reports the success rate of the simulations for ILN when the initial vector of parameters is unusual x_0 = (20.0, -25.0, -20.0, 13.0, 21.0, 30.0, -14.0,-21.0, -13.0, -1.0, 31.0, -8.0, -22.0, 0.0, 4.0, -32.0, 11.0, -11.0, 32.0, -1.5, 12.0,15.2,-11.5, -0.6, 32.7). Note that the optimal parameter of this model is $\hat{x} \approx$ (-1.1,-5.5, 4.9, -7.3, 6.5, -0.64, -2.8, 1.0, -2.97, -1.10, 0.27, -0.52, 0.216, 0.24, 3.21, -1.14, -1.84, -3.28, -2.83, -2.45, 2.51, -2.71, 1.86, 1.37, 1.91), where the optimal log-likelihood value is approximately -3.15, much

higher than the initial log-likelihood value of −275.28.

In Table 4, the TR-MULTI algorithm has a success rate of 50%, a clear dominance over the other algorithms. Simulations fail in Table 4 mostly on the "STEP SIZE" condition (see Table 2), where algorithms stop at points which are very far from the optimal solution. We also observed some failures due to the "TOO MANY ITERATIONS" condition (each estimation was limited to 500 iterations). Interestingly, two failure cases due to the "TOO MANY ITERATIONS" condition had a final log-likelihoods very close to the optimum. These failures belong to the TR-BHHH algorithm.

None of the line search algorithms converged, failing to compute a step size that satisfies the strong Wolfe conditions at the beginning of the optimization process (after only few iterations). This observation suggests that the trust region algorithms are more robust than the line search algorithms.

Algorithms		Successful cases
Trust region	TR-MULTI	5/10
	TR-PRED	3/10
	TR-BUNCH[1]	3/10
	TR-BHHH	0/10
	TR-BFGS	1/10
	TR-SR1	0/10
Line search	LS-BHHH	0/10
	LS-BFGS	0/10
	LS-PRED	0/10

Table 4: Rate of successful simulations for a difficult case (ILN)

In Bunch's approaches, switching from BHHH to a corrected BHHH approximation only occurs once during the optimization process. In Table 5 we report the average number of switching over 10 simulations for our three switching models TR-PRED, TR-MULTI and LS-PRED. In this table we observe a small average number of switches for the LS-PRED, which means that LS-PRED often uses BHHH during the all the optimization process or switched one or two times to a corrected BHHH and then uses a fixed approximation scheme until convergence. The contrast with the number of switches in the trust-region methods can be partly explained as the possibility for switching is considered once per iteration, but the trust region methods typically requires more, but cheaper, iterations than line search techniques. Moreover, the additional efforts made at each iteration of the line search algorithm to satisfy the Wolfe conditions provide a step that is often more efficient for the model under consideration, limiting the potential for other models to provide a higher func-

tion decrease at the new iterate. This suggests that the trust region approach is more suited for switching strategies.

Algorithms		SP2	IN	ILN	PS	PSL
Trust region	TR-PRED	5.3	4.7	18.4	6.4	5.7
	TR-MULTI	6.5	7.8	11.5	8.4	7.7
Line search	LS-PRED	1.4	1.0	0.9	1.0	1.0

Table 5: Average number of switches per estimation in the switching models

8 Conclusion

In this paper, we have reviewed standard trust region and line search algorithms for maximum likelihood estimation, with emphasis on the use of Hessian approximation methods to determine the step at each iteration. We have explored the possibility of switching between various Hessian approximations throughout the optimization process. In particular, we propose a predictive approach, aiming to determine the most suited model between several quadratic approximations at each iteration. This approach does not require any new computation of the objective function. We have also proposed the multi-subproblems model which is based on the fact that, at each iteration of a trust region algorithm, we can solve more than one sub-problem to better determine a step. This approach however requires additional evaluations of the objective function which depends on the size of the set of Hessian approximations considered at each iteration.

We have applied our algorithms to mixed logit and logit based route choice models based on real data sets. The predictive model outperforms the switching approaches proposed by Bunch (1987) as well as the classical optimization algorithms. The multi-subproblems requires large numbers of function evaluations but it has the highest successful rates when solving a complex mixed logit model with an unusual starting point.

In future research we plan to extend further the switching models to use information from several iterations to improve the accurateness of the switching strategies, and to combine this approach with adaptive sampling strategies for mixed logit models (Bastin et al., 2006). We also emphasize that the proposed algorithms can be applied to least square problems too, extending the method proposed by (Dennis Jr et al., 1981). Moreover, we plan to extend the switching algorithms to other classes of non-linear optimization problems.

Acknowledgements

The authors would like to thank NSERC for partial support of this research.

References

F. Bastin, C. Cirillo, and Ph. L. Toint. An adaptive Monte Carlo algorithm for computing mixed logit estimators. *Computational Management Science*, 3(1):55–79, 2006.

F. Bastin, C. Cirillo, and Ph. L. Toint. Estimating non-parametric random utility models, with an application to the value of time in heterogeneous populations. *Transportation Science*, 44(4):537–549, 2010.

M. Ben-Akiva and M. Bierlaire. Discrete choice methods and their applications to short-term travel decisions. In R. W. Hall, editor, *Handbook of Transportation Science*, pages 5–34. Kluwer, Norwell, MA, USA, 1999.

M. Ben-Akiva and S. R. Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand*. The MIT Press, Cambridge, MA, USA, 1985.

E. K. Berndt, B. H. Hall, R. E. Hall, and J. A. Hausman. Estimation and inference in nonlinear structural models. *Annals of Economic and Social Measurement*, 3/4: 653–665, 1974.

C. G. Broyden. The convergence of a class of double-rank minimization algorithms 1. General considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970.

D. S. Bunch. Maximum likelihood estimation of probabilistic choice models. *SIAM Journal on Scientific and Statistical Computing*, 8(1):56–70, 1987.

C. Cirillo and R. Xu. Forecasting cybercar use for airport ground access: A case study at BWI (Baltimore Washington International Airport). *Journal of Urban Planning and Development*, 136(3):186–194, 2010.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Convergence of quasi-Newton matrices generated by the symmetric rank one update. *Mathematical Programming*, 50(2): 177–196, 1991.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. SIAM, Philadelphia, PA, USA, 2000.

J. E. Dennis Jr, D. Gay, and R. E. Welsch. An adaptive nonlinear least-squares algorithm. *ACM Transactions on Mathematical Software*, 7(3):348–368, 1981.

Electric Power Research Institute. Methodology for predicting the demand for new electricity-using goods. Final Report EA-593, Project 488-1, Electric Power Research Institute, Palo Alto, CA, USA, 1977.

R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317–322, 1970.

M. Fosgerau, E. Frejinger, and A. Karlstrom. A link based network route choice model with unrestricted choice set. *Transportation Research Part B*, 56:70–80, 2013.

E. Frejinger. Random sampling of alternatives in a route choice context. In *Proceedings of the European Transport Conference*, Leiden, The Netherlands, 2007.

E. Frejinger, M. Bierlaire, and M. Ben-Akiva. Sampling of alternatives for route choice modeling. *Transportation Research Part B*, 43(10):984–994, 2009.

D. Goldfarb. A family of variable metric updates derived by variational means. *Mathematics of Computation*, 24(109):23–26, 1970.

T. Mai, E. Frejinger, and F. Bastin. A misspecification test for logit based route choice models. Technical Report 2014-32, CIRRELT, Montreal, QC, Canada, 2014.

D. McFadden. Conditional logit analysis of qualitative choice behaviour. In P. Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press New York, New York, NY, USA, 1973.

D. L. McFadden. Modelling the choice of residential location. In A. K. et al., editor, *Spatial Interaction Theory and Residential Location*, pages 75–96. North Holland, Amsterdam, The Netherlands, 1978.

D. L. McFadden. Econometric models of probabilistic choice. In C. F. Manski and D. L. McFadden, editors, *Structural Analysis of Discrete Data with Econometric Applications*, pages 198–272. MIT Press, Cambridge, MA, USA, 1981.

D. L. McFadden and K. Train. Mixed MNL models for discrete response. *Journal of Applied Econometrics*, 15(5):447–270, 2000.

D. Munger, P. L'Ecuyer, F. Bastin, C. Cirillo, and B. Tuffin. Estimation of the mixed logit likelihood function by randomized quasi-monte carlo. *Transportation Research Part B*, 46(2):305–320, 2012.

W. K. Newey and D. McFadden. Large sample estimation and hypothesis testing. In R. Engle and D. McFadden, editors, *Handbook of Econometrics*, volume IV, chapter 36, pages 2111–2245. Elsevier, Amsterdam, The Netherlands, 1986.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, 2nd edition, 2006.

F. Öztoprak and Ş. İ. Birbil. A symmetric rank-one quasi-newton line-search method using negative curvature directions. *Optimization Methods and Software*, 26(3): 455–486, 2011.

P. K.-H. Phua and R. Setiono. Combined quasi-Newton updates for unconstrained optimization. Technical Report TR41/92, National University of Singapore, Department of Information Systems and Computer Science, 1992.

D. Revelt and K. Train. Mixed logit with repeated choices. *Review of Economics and Statistics: Households' Choices of Applicance Efficiency Effect*, 80(4), 1998.

D. F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, 24(111):647–656, 1970.

T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.

Ph. L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. In I. S. Duff, editor, *Sparse Matrices and Their Uses*, pages 57–88. Academic Press, London, England, 1981.

K. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, New York, NY, USA, second edition, 2009.

P. Vovsha and S. Bekhor. Link-nested logit model of route choice Overcoming route overlapping problem. *Transportation Research Record*, 1645:133–142, 1998.