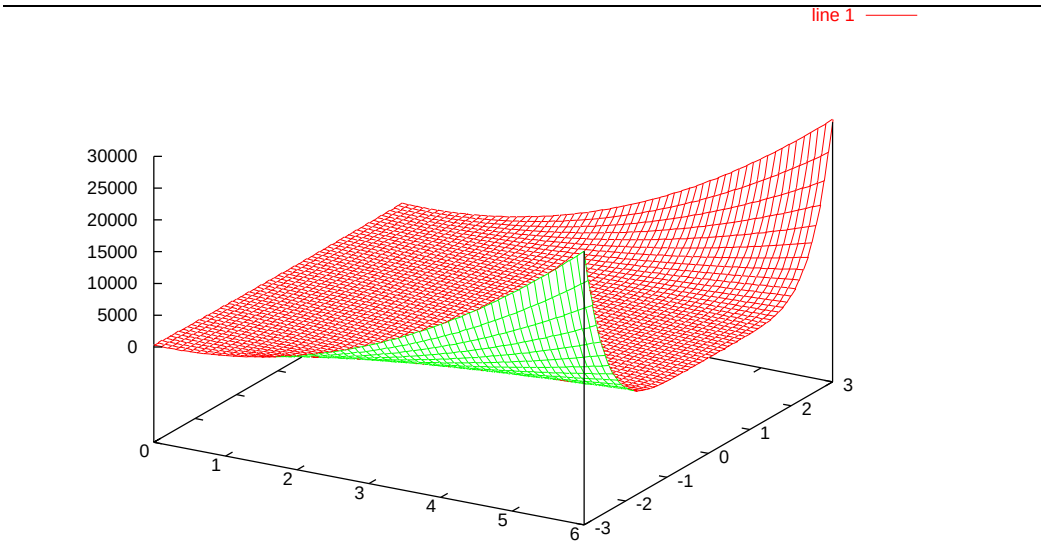

Algorithmic developments for the estimation of advanced discrete choice models

Michel Bierlaire, EPFL
Michaël Thémans, EPFL

Conference paper STRC 2005
Session Transport Modelling I

STRC | **5th Swiss Transport Research Conference**
Monte Verità / Ascona, March 9-11, 2005

Algorithmic developments for the estimation of advanced discrete choice models

Michel Bierlaire and Michaël Thémans
Institute of Mathematics, EPFL, Lausanne
Email: michel.bierlaire@epfl.ch, michael.themans@epfl.ch

Abstract

Discrete choice models have received a great amount of attention in the last years. Recent advances have proposed new models in the Generalized Extreme Value (GEV) family, mixed Logit models and even mixed GEV models. Estimating those models becomes more and more problematic. The objective function becomes highly nonlinear and non concave and the complexity of the model forces to impose constraints (maybe nonlinear) on the parameters in order to obtain meaningful values or to overcome model overspecification which can lead to singularity in the objective function. State-of-the-art algorithms can no longer be applied and specific optimization algorithms must be developed for the estimation of advanced discrete choice models. In particular, we need optimization algorithms able to deal with singularity (possibly nonlinear). Our first step toward this goal is to investigate the case of unconstrained optimization when an affine singularity arises in the objective function. First, in order to get a robust method in the presence of a singularity, we propose to perform an eigen-structure analysis on the second derivatives matrix of the objective function which allows us to characterize the subspace in which the singularity lies. Second, once the singularity has been properly identified, we fix this singularity by adding constraints describing it in order to better guide the algorithm toward a local solution of the optimization problem. The difficulty here is that the identification of the singularity is an iterative process taking place within the optimization algorithm. Therefore, those constraints must be included "on-the-fly" in the optimization problem. This second part is achieved using a multidimensional filter which is an algorithmic tool coming from multi-criteria optimization. In our context, we want to maximize the log-likelihood function but also to satisfy the constraints associated with the singularity. In this paper, we present preliminary numerical

results with the current version of our algorithm designed to solve singular unconstrained optimization problems. We show that our methods should significantly decrease the model estimation time. We also discuss the future modifications to generalize our algorithm to deal with singular constrained optimization.

Keywords: Nonlinear optimization, Maximum Likelihood Estimation, Discrete Choice Models, Singularity, Model Overspecification
Swiss Transport Research Conference – STRC 2005 – Monte Verita

1 Introduction

The theoretical foundations of discrete choice models (and more specifically, random utility models) had already been defined in the seventies (Ben-Akiva, 1973, Williams, 1977, McFadden, 1978) with the Multinomial Logit model, the Multinomial Probit model, the Nested Logit model, and the Generalized Extreme Value model. However, only the Multinomial logit model and the Nested Logit model have been intensively used by practitioners during almost three decades. These models are relatively easy to estimate, as their associated log-likelihood function has nice properties (globally concave for the Multinomial Logit model, concave in a subspace of parameters for the Nested logit model). Therefore, the use of the classical Newton-Raphson optimization algorithm is most of the time appropriate. However, in the presence of poorly significant parameters, the speed of convergence can be very slow.

Recent advances in discrete choice models are following two complementary tracks. Firstly, more "logit-like" models within the Generalized Extreme Value family have been proposed and used (see, for instance, Bierlaire, 2002 and Daly and Bierlaire, 2003). Secondly, the increasing power of computers has motivated the use of Mixed Logit models, where the normal distribution of some parameters requires simulation methods to compute the probability model (McFadden and Train, 1997, Bhat, 2001). Actually, GEV models with mixed distribution start to be proposed as well in the literature (see Hess et al., 2004a and Hess et al., 2004b).

Estimating those models, that is computing the maximum log-likelihood, becomes more and more problematic. Firstly, the objective function becomes highly nonlinear and non concave. Secondly, the computational cost of evaluating the objective function and its derivatives becomes significantly high. Thirdly, the complexity of the model

often requires constraints on the parameters, in order to obtain meaningful values, or to overcome model overspecification. This last issue in the maximum likelihood estimation problem corresponds to a singularity in the likelihood function to be maximized. Classical unconstrained optimization algorithms can no longer be applied and we thus need optimization algorithms able to deal with singularity and capable of handling non trivial (possibly highly nonlinear) constraints on the variables.

In this paper, we are interested in dealing with the singularity which can arise in the maximum likelihood estimation. The cause of the singularity can be multiple, for instance:

- The theoretical model contains too many parameters and not all of them are identifiable (in this case we speak about model overspecification).
- The utility functions contain irrelevant attributes.
- The specification of the utility functions contains more parameters than the data allows to estimate, due to a lack of variability.

In the first case, the singularity is structural in the sense that it is due to the theoretical model used. In the last two cases, the source of the singularity comes from a poor model specification by the modeler, which frequently happens during the model development phase.

We now illustrate a very simple example of singularity in the maximum likelihood estimation by considering an obviously overspecified model in a mode choice context. Suppose that the choice set is $\mathcal{C} = \{\text{train}, \text{car}\}$ and that we have a sample composed of N observed choices. We define the utility functions of each alternative as follows

$$\begin{aligned} U_{\text{t}} &= \beta_1 \text{cost}_{\text{t}} + \beta_2 \text{time}_{\text{t}} + \beta_3 \text{income} + \varepsilon_{\text{t}}, \\ U_{\text{c}} &= \beta_1 \text{cost}_{\text{c}} + \beta_2 \text{time}_{\text{c}} + \beta_3 \text{income} + \varepsilon_{\text{c}}. \end{aligned}$$

We do not impose constraints on the β_i's parameters. Defining the log-likelihood of the sample by

$$\bar{\mathcal{L}}(\beta) = \sum_{n=1}^{N} (y_{\text{train},n} \log P_n(\text{train}|\beta) + y_{\text{car},n} \log P_n(\text{car}|\beta)),$$

with

$$y_{\text{train},n} = \begin{cases} 1 & \text{if the } n\text{-th observed choice is train}, \\ 0 & \text{otherwise}, \end{cases}$$

and

$$y_{\text{car},n} = \begin{cases} 1 & \text{if the } n\text{-th observed choice is car,} \\ 0 & \text{otherwise,} \end{cases}$$

where the probabilities of observed choices depend on the discrete choice model we consider (for instance a Multinomial Logit model) and noting $\beta = (\beta_1, \beta_2, \beta_3)^T$, the estimation process corresponds to solving the following unconstrained nonlinear optimization problem

$$\max_{\beta \in \mathbb{R}^3} \bar{\mathcal{L}}(\beta).$$

Given the above model specification, we easily see that

$$P_n(\text{train}|\beta) = P_n(\text{train}|\beta + \alpha e_3) \quad \forall \alpha \in \mathbb{R},$$

and, equivalently, that

$$P_n(\text{car}|\beta) = P_n(\text{car}|\beta + \alpha e_3) \quad \forall \alpha \in \mathbb{R},$$

where e_3 is the third canonical vector in $\mathbb{R}^3$.

In this case, for each observed choice n in the sample, we have that

$$\frac{\partial P_n(\text{train})}{\partial \beta_3} = \frac{\partial P_n(\text{car})}{\partial \beta_3} = 0,$$

and, as a consequence, we obtain that

$$\frac{\partial \bar{\mathcal{L}}}{\partial \beta_3} = 0.$$

It immediately follows that

$$\frac{\partial^2 \bar{\mathcal{L}}}{\partial \beta_3 \partial \beta_i} = \frac{\partial^2 \bar{\mathcal{L}}}{\partial \beta_i \partial \beta_3} = 0 \quad \forall i = 1, \ldots, 3.$$

We conclude that, in this example, the second derivatives matrix of the log-likelihood function, $\nabla^2 \bar{\mathcal{L}}(\beta)$, is singular of dimension 1 for all $\beta \in \mathbb{R}^3$. In particular, $\nabla^2 \bar{\mathcal{L}}(\beta^*)$, where β^* is a local solution, is singular.

The most simple examples of structurally unidentifiable models are the variance parameter, and the Alternative Specific Constants (ASCs) in the Multinomial Logit model (MNL) model. The variance cannot be identified, and only J-1 ASCs can be identified, in a model with J alternatives. In this context, a detailed analysis of the overspecification due to ASCs is provided by Bierlaire et al. (1997). In this case, the singularity can be easily fixed directly in the model specification, and no specialized

optimization algorithm is required to solve the maximum log-likelihood estimation problem.

A singularity in the log-likelihood function has two main drawbacks. Firstly, the convergence of the estimation process will be slower. In the case of an unconstrained optimization problem (namely maximizing the log-likelihood function without constraints on the parameters), a singularity means that the second derivatives matrix of the objective function is not invertible, violating one of the main assumptions underlying the convergence theory of Newton-like methods. In this context, if the second derivatives matrix $\nabla^2\bar{\mathcal{L}}(\beta^*)$ is non-singular at a local minimizer β^*, Newton's method is known to exhibit a quadratic rate of local convergence to β^*. But one shortcoming of Newton-like methods for unconstrained optimization is that they do not converge quickly if the Hessian at the minimizer, $\nabla^2\bar{\mathcal{L}}(\beta^*)$, is singular. Griewank and Osborne (1983) have shown that in this case, the iterates produced are at best linearly convergent (even if the second derivatives matrix is non-singular at all iterates). Furthermore, when solving singular problems, standard methods can encounter numerical problems. Secondly, the variance-covariance matrix of the estimates cannot be obtained from the inversion of $\nabla^2\bar{\mathcal{L}}(\beta^*)$. As a consequence, statistical tests of these estimates are no more available, meaning that it is not possible to assess the quality of the calibrated model.

It is interesting to note that the possible singularity of the maximum log-likelihood problem has almost not been addressed in the literature. However, Walker (2001) has shown that identification issues appear with simple models, and cannot always be easily addressed. Namely, not all standard errors can be estimated in an Error Component model. But the choice of the standard error to fix to 0 is not known a priori, and an overspecified model must be estimated first. Fixing the singularity in a more complex model can also be problematic.

In non trivial cases, we need a specialized optimization algorithm which is able to detect a singularity during the course of the algorithm (that is during the estimation process) and subsequently fix it. This is the scope of this paper. Our first step toward a specialized optimisation algorithm designed to estimate advanced discrete choice models is the development of two algorithms able to solve unconstrained maximum likelihood estimation problems which contain singularities.

The specifications of a singular unconstrained problem as well as a technique to handle the singularity is presented in Section 2. Two algorithms containing modifications designed to solve singular problems efficiently are described in Section 3. Preliminary results of these methods applied to singular problems are then presented

in Section 4. Before concluding in Section 6, we give some tracks we will follow in our future research in Section 5.

2 Ideas

We present the main features of an algorithm which is designed to solve unconstrained singular problems. In this paper, we focus on the case of a linear singularity.

To be consistent with the nonlinear optimization literature, we define $x = \beta$ and $f(x) = -\bar{\mathcal{L}}(\beta)$. Therefore the maximum log-likelihood estimation problem is written

$$\begin{cases} \min f(x) \\ x \in \mathbb{R}^n \end{cases} \iff \begin{cases} \max \bar{\mathcal{L}}(\beta) \\ \beta \in \mathbb{R}^n \end{cases}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be twice continuously differentiable. We assume that the problem is singular, that is, $\exists$ A and d such that

$$f(x^*) = f(x^* + Ad),$$

where :

- x^* is a local minimum of the problem,
- $d \in \mathbb{R}^m$,
- $A \in \mathbb{R}^{n \times m}$ is full column-rank with $m < n$.

A represents the eigen-subspace associated with the null eigenvalues of the second derivatives matrix at x^*, $\nabla^2 f(x^*)$. The range of A, $\mathrm{Im}(A)$, characterizes the subspace of $\mathbb{R}^n$ in which the singularity lies at the local solution. Invoking the fundamental theorem of linear algebra, we know that the subspace orthogonal to $\mathrm{Im}(A)$, $\mathrm{Im}(A)^\perp$, is equivalent to $\ker(A^T)$.

In this case, the problem $\min_{x \in \mathbb{R}^n} f(x)$ is equivalent to

$$\begin{cases} \min f(x) \\ \text{s.c. } A^T x = 0. \end{cases}$$

The difficulty is that A is unknown before the optimization process starts. We would like to detect the degradation of the algorithm's convergence when applied to this kind of problem. In general, this will occur when the iterates reach a vicinity of the set of solutions.

Given that, the main idea is to perform an eigen-structure analysis of the second derivatives matrix $\nabla^2 f(x_k)$ at the current iterate. The eigen-subspace associated with eigenvalues close to zero would be a good approximation for the range of A, provided that x_k is close enough to the set of solutions. In practice, performing a QR factorization at each iteration is cumbersome.

In order to identify the singular subspace during the course of the algorithm, we have developed a generalization of the inverse iteration method (see, for instance, Golub and Loan, 1996). This iterative process, designed to be applied on a symmetric matrix, allows the identification of the closest eigenvalue (in modulus) to a given shift as well as the associated eigenvector. We generalize this method in order to compute higher-dimensional invariant subspaces. Indeed, we are able to approximate the subspace associated with the r closest eigenvalues (in modulus) to a given shift, with $1 \leq r \leq n$ where n is the dimension of the square matrix we are investigating.

The main steps of the identification process are:

- A matrix $H \in \mathbb{R}^{n \times n}$ and r, a chosen integer satisfying $1 \leq r \leq n$, are given.
- We construct the new matrix $\bar{H} = (H - \lambda I_{n \times n})^{-1}$ where λ is called the shift.
- Using our generalization of the inverse iteration, we then compute the matrix $Q \in \mathbb{R}^{n \times r}$ such that $\mathrm{Im}(Q)$ approximates the dominant invariant subspace of dimension r of $\bar{H}$, denoted $\mathcal{D}_r(\bar{H})$.

It allows us to get an approximation of $\mathcal{D}_r(\bar{H})$, namely the subspace associated with the r largest eigenvalues in modulus of $\bar{H}$. With regard to H, $\mathcal{D}_r(\bar{H})$ represents the subspace associated with the r closest eigenvalues (in modulus) to the given shift λ.

For our purpose, we obviously fix the shift to a small value, say 10^{-10}. Note that we do not choose 0 for numerical reasons. Recall that we want to identify the subspace associated with eigenvalues close to 0, or at least less in modulus than a given threshold.

At each iteration of our optimization algorithm, we perform the above eigen-structure analysis on $H = \nabla^2 f(x_k)$. If we identify that the smallest eigenvalues in modulus are too close to 0, it means that the iterates are reaching the vicinity of the set of solutions. From that moment, we need to fix the singularity, keeping the iterates in the orthogonal subspace to the one in which the singularity lies. More precisely, we add constraints describing the singularity. If Q_k is the current approximation of the

eigen-subspace associated with the singularity, we define the constraint $Q_k^T x = 0$ in order to better guide the algorithm toward a local solution of the optimization problem, keeping the iterates in the subspace in which the information about curvature is relevant.

The way we include this type of constraint during the course of the algorithm will be described in details in the next section where we present two algorithms to solve unconstrained singular problems. In the first algorithm we use a penalty approach by adding a penalty to the violation of the constraint in the minimization subproblem we solve at each iteration of the algorithm. We consider a penalty term of the form $\frac{1}{2}c\|Q_k^T x\|^2$ where c is the penalty parameter which determines the weight of this term in the minimization. In the second algorithm, we keep this penalty term in the minimization subproblem but we also make use of the constraint violation $\|Q_k^T x\|$ as a measure of progress toward the solution using a multidimensional filter. In our context, we want to minimize the objective function $f(x)$ but also satisfy the constraints associated with the singularity.

3 Algorithms

3.1 Trust-region based algorithm

The first algorithm we present is inspired by the Basic Trust-Region framework presented in Conn et al. (2000) to solve unconstrained optimization problems of the type

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

where $f(x)$, the objective function, is a real-valued twice-continuously differentiable function, which we assume is bounded below. We present an iterative numerical procedure in which the objective function is approximated in a suitable neighborhood of the current iterate (we call it the trust-region) by a model which is easier to handle than $f(x)$.

A trust-region algorithm works as follows. At each iterate x_k, we first define a model $m_k(x)$ whose purpose is to approximate the objective function in a suitable neighborhood of x_k, called the trust-region. The trust-region is defined as the following set of points

$$\mathcal{B}_k = \{x \in \mathbb{R}^n \mid \|x - x_k\| \leq \Delta_k\},$$

where Δ_k is called the trust-region radius, and where $\| \ . \ \|$ denotes the classical Euclidean norm. Given this model, we look for a trial step s_k such that the trial point

defined as $x_k + s_k$ reduces the model while satisfying the constraint $\|s_k\| \leq \Delta_k$. Having computed this trial step, we now compute the objective function at $x_k + s_k$ and we compare this value with the value predicted by the model, that is $m_k(x_k + s_k)$. If the reduction predicted by the model is realized also in the objective function, the trial point is accepted to be the next iterate and the trust-region is kept the same or even expanded, depending on the quality of the reduction in the objective function. If it appears that the reduction in the model is a poor predictor of the actual reduction in the objective function, the trial point is rejected and the trust-region is reduced, hoping that we will get a better model of the objective function in a smaller neighborhood of the current iterate.

More formally, a trust-region based algorithm can be described as follows.

Step 0: Initialization. An initial point x_0, an initial trust-region radius Δ_0 and a tolerance ε are given. The constants η_1, η_2, α_1, α_2 and α_3 are also given and they satisfy

$$0 < \eta_1 \leq \eta_2 < 1 \quad \text{and} \quad 0 < \alpha_3 \leq \alpha_2 \leq 1 \leq \alpha_1.$$

Compute $f(x_0)$ and set $k = 0$.

Step 1: Model definition. Define a model m_k in $\mathcal{B}_k$.

Step 2: Step computation. Compute a step s_k that sufficiently reduces the model m_k and such that $x_k + s_k \in \mathcal{B}_k$. This step is also called the trust-region subproblem because we solve approximately the following problem

$$\begin{cases} \min m_k(x_k + s) \\ \text{s.c. } \|s\| \leq \Delta_k. \end{cases}$$

Step 3: Acceptation of the trial point. Compute $f(x_k + s_k)$ and define

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

If $\rho_k \geq \eta_1$, then define $x_{k+1} = x_k + s_k$; otherwise define $x_{k+1} = x_k$.

Step 4: Trust-region radius update. Set

$$\Delta_{k+1} = \begin{cases} \max(\alpha_1\|s_k\|, \Delta_k) & \text{if } \rho_k \geq \eta_2, \\ \alpha_2\Delta_k & \text{if } \rho_k \in [\eta_1, \eta_2), \\ \alpha_3\|s_k\| & \text{if } \rho_k < \eta_1. \end{cases}$$

Increment k by 1. If $\|\nabla f(x_k)\| \geq \varepsilon$ go to Step 1; otherwise stop.

In the literature of nonlinear optimization, we often use a quadratic model of the form

$$m_k(x_k + s) = f(x_k) + g_k^T s + \frac{1}{2} s^T H_k s,$$

where

$$m_k(x_k) = f(x_k) \text{ and } g_k = \nabla f(x_k),$$

and where H_k is a symmetric aproximation of $\nabla^2 f(x_k)$ using finite differences. A secant approximation of the Hessian matrix based on previous iterates could also be used as H_k.

In our context, remember that once we have identified an approximation of the subspace in which a singularity lies, we want to fix it, constraining the subsequent iterates to be in the orthogonal subspace.

At iteration $k+1$, given x_k and H_k, we perform on H_k the eigen-structure analysis described in the previous section, and if we detect a subspace Q_k associated with eigenvalues close to 0, we consider in the trust-region subproblem the following model

$$\widehat{m}_k(x_k + s) = f(x_k) + g_k^T s + \frac{1}{2} s^T H_k s + \frac{1}{2} c \|Q_k^T s\|,$$

where $c > 0$ is the penalty parameter which gives the weight associated with the constraint violation. The second derivatives matrix of this model is given by

$$\nabla_{xx} \widehat{m}_k(x_k) = H_k + c\, Q_k Q_k^T.$$

It means that we add a multiple of a positive definite matrix to the second derivatives matrix of f when it is becoming close to a singular matrix. The idea of this bending strategy is thus to add some curvature in order to overcome the flatness of the objective function.

Actually, this model can be viewed as a perturbation of the classical second-order model defined above. It also satisfies the underlying assumptions on the model in a trust-region framework in order to guarantee the convergence to a local minimum of the optimization problem.

Note that the trust-region subproblem (independently from the fact that we have or not the additional term in it) is solved using a Truncated Conjugate Gradient method (see, for instance, Toint, 1981 or Steihaug, 1983). In this case, it can be shown under mild assumptions that the overall trust-region algorithm will converge superlinearly to a solution of the optimization problem.

3.2 Filter trust-region based algorithm

With this algorithm, we add to the trust-region framework an additional concept called the filter. The filter technique has been introduced by Fletcher and Leyffer (2002) in the context of constrained optimization. The filter concept comes from multi-criteria optimization and allows to deal with different objectives to measure progress toward the solution of a problem. Fletcher and Leyffer (2002) defined a 2-dimensional filter associated with the two objectives of constrained optimization, namely minimizing the objective function while satisfying the constraints. Later Gould et al. (2005) generalized the concept by using a multidimensional filter to solve systems of nonlinear equations as well as nonlinear least-squares. A multidimensional filter is also used in Gould et al. (2004) in the context of unconstrained optimization. The advantage of the filter is the increased flexibility in the optimization algorithm to accept new iterates, and consequently, the potentially fast convergence.

We now present our second algorithm to solve unconstrained optimization problems based on the algorithm described in Gould et al. (2004) and then we present the modifications necessary to deal with singularity issues.

We extend trust-region methods by introducing a multidimensional filter technique, whose aim is to encourage the convergence of iterates to a first-order critical point, by driving each component of the gradient of the objective function $\nabla f(x) = g(x) = (g_1(x), \ldots, g_n(x))^T$ to zero.

In comparison with the trust-region algorithm presented in the previous section, we now consider a filter mecanism in order to potentially accept the trial point $x_k + s_k$ more often. The notion of filter is based on the concept of dominance. In our case, we will say that an iterate x_1 is dominated by an iterate x_2 when

$$|g_i(x_2)| \leq |g_i(x_1)| \quad \forall i = 1, \ldots, n.$$

So, if we want to focus our attention on convergence to first-order critical points, the iterate x_1 is of no real interest because the iterate x_2 is better than x_1 with regard to each component of the gradient. Using this concept, all we have to do is to remember all non-dominated iterates. We do it by using the so-called filter structure. We define a multidimensional filter $\mathcal{F}$ as a list of n-tuples $(g_{k,1}, \ldots, g_{k,n})$ with $g_{k,i} = g_i(x_k)$ such that, if $g_k \in \mathcal{F}$, then we have that

$$|g_{k,j}| < |g_{l,j}| \text{ for at least one } j \in \{1, \ldots, n\}$$

$\forall\ g_l \in \mathcal{F}$. It means that each point in the filter is not dominated by any other point in the filter.

In a filter method, we accept a new trial point $x_k + s_k$ if it is not dominated by any other point in the filter. However, from an algorithmic point of view, we do not want to accept a trial point which is arbitrarily close to a point in the filter. This is why we slightly strengthen the acceptability test and we thus say that a trial point $x_k^+ = x_k + s_k$ is acceptable for the filter $\mathcal{F}$ if

$$\forall\, g_l \in \mathcal{F} \quad \exists\, j \in \{1, \ldots, n\} \text{ such that } |g_j(x_k^+)| \leq (1 - \gamma_\theta)|g_{l,j}|,$$

where γ_θ is a small positive constant. If an iterate x_k is acceptable for the filter and if we decide to add it to the filter, we remove all dominated entries $g_l \in \mathcal{F}$ such that $|g_{l,j}| > |g_{k,j}| \; \forall\, j \in \{1, \ldots, n\}$.

Remember that the second algorithm we present is designed to solve unconstrained optimization problems. However, the filter mechanism only guide the iterates toward a zero gradient. It is adequate for convex problems where a zero gradient is both necessary and sufficient condition for second-order optimality but it may be inappropriate for nonconvex ones. For example, it might prevent progress away from a saddle point where, in this case, an increase in the components of the gradient is acceptable. We therefore adapt the filter mechanism presented above by a reset to zero of the filter after an iteration for which a sufficient decrease in the objective function is achieved using a model m_k being nonconvex. In this case, we also define an upper bound on the acceptable objective function values in order to keep a monotone algorithm in term of objective function value.

We now present an algorithm which combines these ideas with the trust-region algorithm presented in the previous section. Basically, the filter plays the major role in ensuring the convergence when convexity is present in the model, while falling back on the classical trust-region algorithm if negative curvature is detected (during the resolution of the trust-region subproblem).

More formally, a filter-trust-region based algorithm can be described as follows.

Step 0: Initialization. An initial point x_0, an initial trust-region radius Δ_0 and a tolerance ε are given. The constants γ_θ, η_1, η_2, α_1, α_2 and α_3 are also given and they satisfy

$$0 < \eta_1 \leq \eta_2 < 1 \quad \text{and} \quad 0 < \alpha_3 \leq \alpha_2 \leq 1 \leq \alpha_1.$$

Compute $f(x_0)$ and $g(x_0)$ and set $k = 0$. Initialize the filter $\mathcal{F}$ to the empty set. Define an initial upper bound $f_{sup} \geq f(x_0)$. Define the flag nonconvex unset.

Step 1: Model definition. Define a model m_k in $\mathcal{B}_k$.

Step 2: Step computation. Compute a step s_k that sufficiently reduces the model m_k and such that $x_k + s_k \in \mathcal{B}_k$. If m_k is detected to be nonconvex, set nonconvex; otherwise unset it. Compute $x_k^+ = x_k + s_k$.

Step 3. Compute $f(x_k^+)$ and define

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

If $f(x_k^+) > f_{sup}$, then define $x_{k+1} = x_k$ and go to Step 5.

Step 4: Acceptation of the trial point. Compute $g_k^+ = g(x_k^+)$.

- If x_k^+ is acceptable for the filter $\mathcal{F}$ and nonconvex is unset
  Set $x_{k+1} = x_k^+$ and add g_k^+ to the filter $\mathcal{F}$ if $\rho_k < \eta_1$.
- If x_k^+ is not acceptable for the filter $\mathcal{F}$ or nonconvex is set
  If $\rho_k \geq \eta_1$ then
  Set $x_{k+1} = x_k^+$ and, if nonconvex is set, set $f_{sup} = f(x_{k+1})$ and renitialize the filter $\mathcal{F}$ to the empty set;
  else Set $x_{k+1} = x_k$.

Step 5: Trust-region radius update. Set

$$\Delta_{k+1} = \begin{cases} \max(\alpha_1 \|s_k\|, \Delta_k) & \text{if } \rho_k \geq \eta_2, \\ \alpha_2 \Delta_k & \text{if } \rho_k \in [\eta_1, \eta_2), \\ \alpha_3 \|s_k\| & \text{if } \rho_k < \eta_1. \end{cases}$$

Increment k by 1. If $\|\nabla f(x_k)\| \geq \varepsilon$ go to Step 1; otherwise stop.

This algorithm can be shown to be globally convergent to at least one second-order critical point.

Looking at the phase of the algorithm in which we decide whether a trial point is acceptable to be the next iterate or not, we see that this second algorithm potentially accept the trial point more often than the previous one. Indeed, if the trial point is acceptable for the filter, we move toward this point and if it is not, we look at the quality of the reduction factor ρ_k as in the first algorithm.

We now turn to the modifications we make to handle singularity in the objective function. We actually make two adaptations in this algorithm. The first one is similar

to what we did in the trust-region algorithm. As soon as we detect a singularity in the objective function thanks to the identification process, we do not use the most common quadratic model anymore but we prefer rather using $\widehat{m}_k$, that is

$$\widehat{m}_k(x_k + s) = f(x_k) + g_k^T s + \frac{1}{2} s^T H_k s + \frac{1}{2} c \|Q_k^T s\|,$$

where $c > 0$ is the penalty parameter which gives the weight associated with the violation of the constraint describing the singularity.

This first adaptation is actually a modification of the trust-region subproblem we solve in order to keep the iterates in the subspace in which we have relevant information about the curvature of the objective function.

When a singularity has been identified, the second modification uses the singularity constraint violation to decide whether a trial point is acceptable or not. Indeed, when singularity is detected in the objective function, we not only use our auxiliar model $\widehat{m}_k$ but we also use an auxiliary filter $\widehat{\mathcal{F}}$ in which we add a dimension compared to the filter $\mathcal{F}$ composed of the gradient components. Indeed, for each entry $x \in \mathcal{F}$, we compute $\|Q_k^T x\|$. So we will say that a point if acceptable for the new filter $\widehat{\mathcal{F}}$ if it significantly reduces at least one of the gradient components or the violation of the singularity constraint compared to the previous iterates in the filter.

4 Preliminary numerical results

We present in this section an analysis of the performance of our adaptations to deal with singularity in the context of unconstrained nonlinear optimization. All algorithms and test functions have been implemented with the package Octave (see `http://www.octave.org` or Eaton, 1997) and computations have been done on a desktop equipped with 3GHz CPU in double precision.

The set of test functions has been proposed by More et al. (1981) It is composed, among other things, of 34 unconstrained optimization problems. Most of these problems have a non-singular second derivatives matrix at the local minimum. As we want to perform tests on singular problems, we use the technique proposed by Schnabel and Frank (1984) to modify the problems of More et al. (1981) and create singular optimization problems such that the second derivatives matrix has a rank $n - k$ at the local solution where n is the dimension of the problem and $1 \leq k \leq n$ is the dimension of the singularity. In this paper we will concentrate ourselves on problems having a second-order derivatives matrix of rank $n - 1$ or $n - 2$ as in Schnabel and Chow

(1991). Tests have been actually performed on 31 problems containing a singularity of dimension 1:

- 22 problems with fixed dimension between 2 and 11,
- 3 problems with variable dimension $n = 10, 20, 40$.

We also carried out test on a set of 30 test functions whose second derivatives matrix has rank $n - 2$ at x^*, namely:

- 21 problems with fixed dimension between 3 and 11,
- 3 problems with variable dimension $n = 10, 20, 40$.

For each problem, we have used the starting point given in the original paper of More et al. (1981).

We will consider a total of 4 algorithms, namely the trust-region algorithm, the filter-trust-region algorithm and their corresponding version designed to handle singularity. In the quadratic models we form at each iteration, the approximation H_k of the Hessian at the current iterate is obtained using finite differences. The stopping criterion for all algorithms is a composition of two conditions:gradient close to zero, that is $\|g_k\| \leq 10^{-6}$, and maximum number of iterations fixed to 1000.

The measure of performance is the number of iterations or the CPU time necessary to reach convergence (as defined above). We are presenting the results following the performance profiles analysis method proposed by Dolan and More (2002).

If $f_{\mathsf{p},\mathsf{a}}$ is the performance of algorithm a on problem p, then the *performance ratio* is defined by

$$r_{\mathsf{p},\mathsf{a}} = \frac{f_{\mathsf{p},\mathsf{a}}}{\min_{\mathsf{a}} f_{\mathsf{p},\mathsf{a}}},$$

if algorithm a has converged for problem p, and $r_{\mathsf{p},\mathsf{a}} = r_{\mathtt{fail}}$ otherwise, where $r_{\mathtt{fail}}$ must be strictly larger than any performance ratio. For any given threshold π, the overall performance of algorithm a is given by

$$\rho_{\mathsf{a}}(\pi) = \frac{1}{n_{\mathsf{p}}} \Phi_{\mathsf{a}}(\pi),$$

where n_{p} is the number of problems considered, and $\Phi_{\mathsf{a}}(\pi)$ is the number of problems for which $r_{\mathsf{p},\mathsf{a}} \leq \pi$.

So, in particular, the value $\rho_{\mathsf{a}}(1)$ gives the probability that algorithm a wins over all its competitors. It is a measure of performance. The value $\lim_{\pi \to r_{\mathtt{fail}}} \rho_{\mathsf{a}}(\pi)$ gives the

probability that algorithm a solves a problem and, consequently, provides a measure of robustness of each method.

We first present on Figure 1 the performance in term of number of iterations of the four algorithms on all problems listed above. The results are satisfactory as we see that the two variants designed to handle singularity in the objective function are better in terms of efficiency and robustness. In particular, the modified filter-trust-region algorithm is the best algorithm one more than 70% of the problems and is also the most robust.

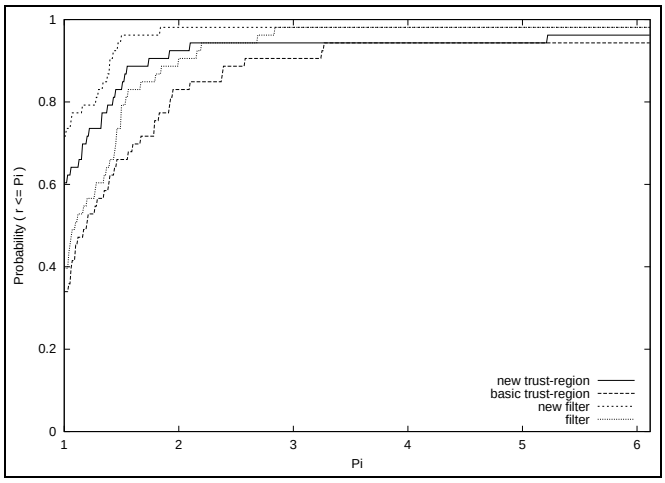

Figure 1: Performance profile for number of iterations - All algorithms

In Figure 2 we show the performance of the two filter-trust-region algorithms with regard to the number of iterations to reach convergence. We see that the algorithm which integrates our adaptations is the best one in around 85% of the time. The two methods are comparable in term of robustness. The Figure 3 presents the performance of these two algorithms in term of CPU time. Despite the computational overhead due to the singularity identification process, we see that the new algorithm takes, on more than 60% of the problems, less time to reach convergence thanks to the smaller number of iterations necessary to conerge to a local solution. On some problems, the new algorithm is 3.5 times faster than the standard one in term of computational time. This is very encouraging for our application purposes. This point is discussed at the end of this section.

Figure 4 and Figure 5 show the performance profiles for the two trust-region algorithms. Same comments can be done on the modified version of the trust-region algorithm as this latter is the fastest and the cheapest one compared to the classical

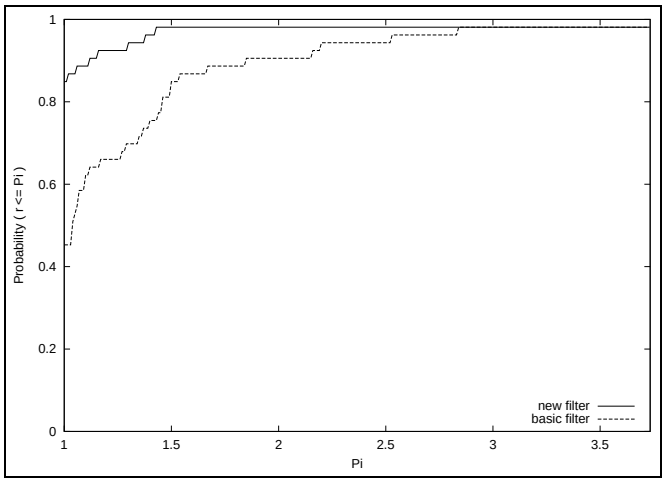

Figure 2: Performance profile for number of iterations - two filter variants

trust-region algorithm. Note that the trust-region variants are not as robust as the filter variants, even if the difference is moderate.

Looking at the performance profiles measuring the CPU time, we see that the overhead in computational costs caused by the identification process is higlhy compensed, in most cases, with the gain in the number of iterations and function evaluations. Nevertheless, most of the problems in our test set are small sized and fuctions are not really expensive to evaluate. This is why we are confident that these algorithmic adaptations should allow a significant gain in the time necessary to estimate advanced discrete choice models. Indeed, in this context, the log-likelihood function can be very expensive to compute, as it can be obtained from simulation tools.

5 Future work and perspectives

Firstly, we will test the algorithmic adaptations presented in this paper on the estimation of real discrete choice models. As discussed in the previous section, we strongly believe that our algorithms will allow an important decrease in the model estimation time when singularity is present in the log-likelihood function.

Secondly, we will investigate in details the singularity identification process. Especially, we will study the convergence rate of the approximations generated by our technique to the real singular subspace. Moreover, we will study the convergence rate of the two algorithms presented here compared to the convergence rate of classical methods to solve unconstrained problems in the presence of singularity.

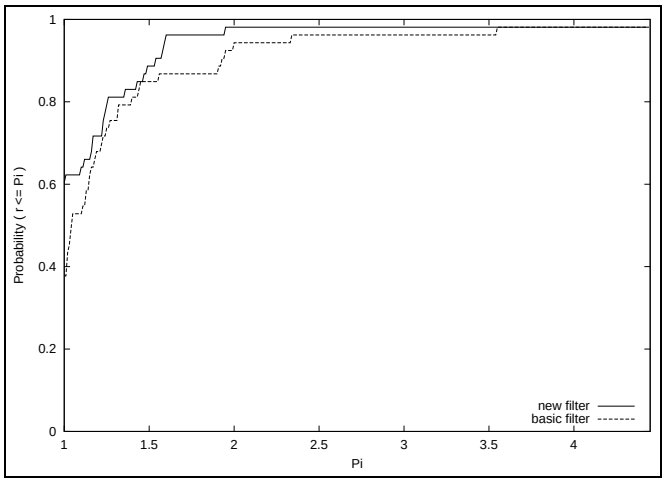

Figure 3: Performance profile for CPU time - two filter variants

Thirdly, this paper deals with singular unconstrained optimization. We definitely want to generalize both theoretical and algorithmic ideas presented in this paper to the case of constrained nonlinear optimization. Our motivation comes from both theoretical and application sides.

From the point of view of applications, we are interested in solving constrained maximum likelihood problems arising when estimating advanced discrete choice models requiring non trivial constraints in order to get meaningful values of parameters as well as to get an identifiable model. In this context, model overspecification is a tricky point and it is necessary to develop specific algorithms to identify the singularity issues and to perform correctly the estimation when non trivial constraints are imposed on the parameters.

From the theoretical point of view, singular constrained optimization is also very interesting. We have seen that a singularity in a unconstrained nonlinear optimization comes from a flat curvature in the vicinity of a local solution, violating one of the major assumptions on the objective function in order to guarantee the fast local convergence of methods. In the constrained case, there may be another source of singularity, namely when a constraint qualification condition is not satisfied (for instance, the assumption of linear independence of the constraints gradients). It is interesting to develop algorithms able to solve efficiently problems for which classical assumptions for convergence of standard methods are violated. Actually, the case of possible violation of standard constraint qualifications is starting to be investigated in the literature of constrained optimization (see, for instance, Wright, 2002, Wright, 2003).

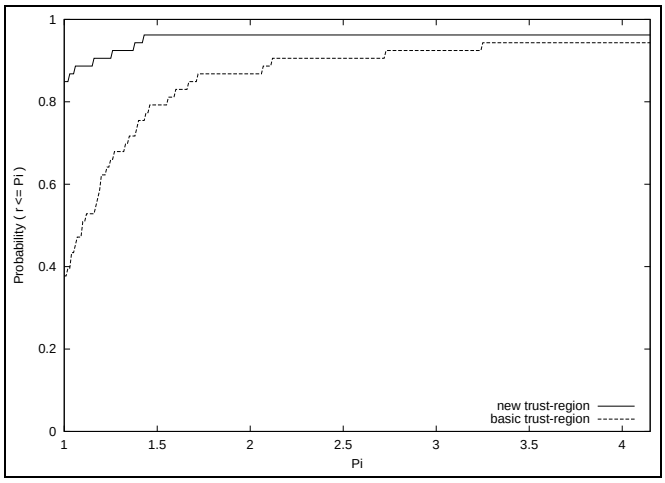

Figure 4: Performance profile for number of iterations - two trust-region variants

Very recently, Izmailov and Solodov (2004) proposed a singular-value decomposition approach in this context. The motivation for considering such irregular cases comes from various problems, where either standard constraints qualifications are inherently violated or constraints tend to be degenerate or nearly (that is numerically) degenerate. Of interest are both theoretical properties of irregular problems as well as convergence of optimization algorithms applied to such problems and, most importantly, possible modifications of the algorithms to improve robustness and efficiency.

6 Conclusion

We propose algorithmic developments in the context of unconstrained nonlinear optimization in order to solve efficiently singular problems. The main contribution is the ability to detect singularity in the objective function during the course of the optimization algorithm as well as the capability to handle adaptative constraints, which allow to fix the singularity, using a penalty approach and/or the filter technique. Preliminary numerical results are encouraging.

It is interesting to consider those pathological problems for which classical assumptions in the theory of nonlinear optimization are violated. Standard methods often exhibit a poor behavior in term of convergence rate on this type of problems. It is also of special interest to propose a method which is more robust faced with the numerical difficulties coming from the singularity present in the problems.

The algorithmic ideas presented in this paper can be used to develop specific opti-

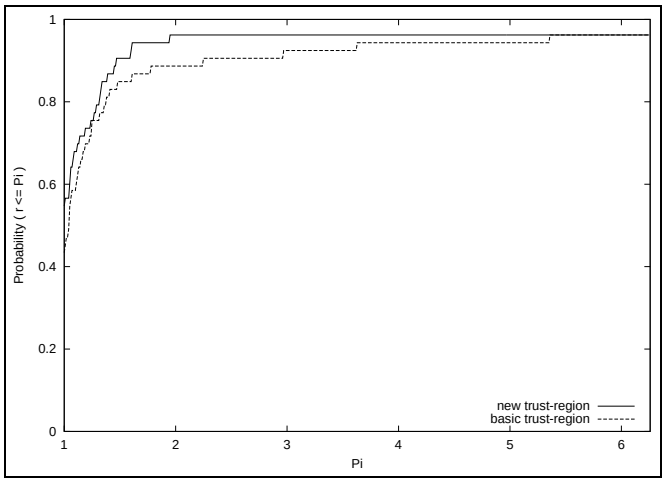

Figure 5: Performance profile for CPU time - two trust-region variants

mization methods designed to estimate advanced discrete choice models. Indeed, the computation of the maximum log-likelihood in the context of dicrete choice models is becoming more and more complicated and requires specific optimization algorithms. In particular, singularity issues arise in the maximum log-likelihood estimation problem. On the one hand, models recently proposed in the literature can be tricky to estimate due to identification issues. Those models contains a lot of theoretical parameters and not all of them can be estimated. On the other hand, it is also of major importance to assist the modeler in the calibration phase by pointing out singularities due to misspecifications.

References

Ben-Akiva, M. E. (1973). *Structure of passenger travel demand models*, PhD thesis, Department of Civil Engineering, MIT, Cambridge, Ma.

Bhat, C. (2001). Quasi-random maximum simulated likelihood estimation of the mixed multinomial logit model, *Transportation Research B* **35**(7): 677–693.

Bierlaire, M. (2002). The network GEV model, Proceedings of the 2nd Swiss Transportation Research Conference, Ascona, Switzerland.

Bierlaire, M., Lotan, T. and Toint, P. L. (1997). On the overspecification of multinomial and nested logit models due to alternative specific constants, *Transportation Science* **31**(4): 363–371.

Conn, A. R., Gould, N. I. M. and Toint, P. L. (2000). *Trust-Region Methods*, Series on Optimization, MPS-SIAM, Philadelphia, USA.

Daly, A. and Bierlaire, M. (2003). A general and operational representation of GEV models, *Technical Report RO-030502*, Institute of Mathematics, Operations Research Group ROSO, EPFL, Lausanne, Switzerland.

Dolan, E. D. and More, J. J. (2002). Benchmarking optimization software with performance profiles, *Mathematical Programming* **91**(2): 201–213.

Eaton, J. W. (1997). GNU *Octave Manual*, Network Theory Limited, Bristol, United Kingdom.

Fletcher, R. and Leyffer, S. (2002). Nonlinear programming without a penalty function, *Mathematical Programming* **91**(2): 239–269.

Golub, G. H. and Loan, C. F. V. (1996). *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, USA.

Gould, N. I. M., Leyffer, S. and Toint, P. L. (2005). A multidimensional filter algorithm for nonlinear equations and nonlineat least-squares, *SIAM Journal on Optimization* **15**(1): 17–38.

Gould, N. I. M., Sainvitu, C. and Toint, P. L. (2004). A filter-trust-region method for unconstrained optimization, *Technical Report TR04-03*, Department of Mathematics, University of Namur, Namur, Belgium.

Griewank, A. O. and Osborne, M. R. (1983). Analysis of newton's method at irregular singularities, *SIAM Journal on Numerical Analysis* **20**(4): 747–773.

Hess, S., Bierlaire, M. and Polak, J. (2004a). Capturing taste heterogeneity and correlation structure with mixed GEV models, *in* A. Alberini and R. Scarpa (eds), *Application of simulation methods in environmental and resource economics*.

Hess, S., Bierlaire, M. and Polak, J. (2004b). Development and application of a mixed cross- nested logit model, *Proceedings of the XXIth European Transport Conference*, Strasbourg.

Izmailov, A. F. and Solodov, M. V. (2004). Newton-type methods for optimization problems without constraint qualifications, *SIAM Journal on Optimization* **15**(1): 210–228.

McFadden, D. (1978). Modelling the choice of residential location, *in* A. K. *et al.* (ed.), *Spatial interaction theory and residential location*, pp. 75–96.

McFadden, D. and Train, K. (1997). Mixed multinomial logit models for discrete response, *Technical report*, University of California, Berkeley, Ca.

More, J. J., Garbow, B. S. and Hillstrom, K. E. (1981). Testing unconstrained optimization software, *ACM Transactions on Mathematical Software* **7**: 17–41.

Schnabel, R. B. and Chow, T. (1991). Tensor methods for unconstrained optimization using second derivatives, *SIAM Journal on Optimization* **1**(3): 293–315.

Schnabel, R. B. and Frank, P. (1984). Tensor methods for nonlinear equations, *SIAM Journal on Numerical Analysis* **21**(5): 815–843.

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

Toint, P. L. (1981). Towards an efficient sparsity exploiting newton method for minimization, *in* I. S. Duff (ed.), *Sparse Matrices and Their Uses*, pp. 57–88.

Walker, J. L. (2001). *Extended discrete choice models: integrated framework, flexible error structures, and latent variables*, PhD thesis, Massachusetts Institute of Technology, Cambridge, Ma.

Williams, H. (1977). On the formation of travel demand models and economic measures of user benefit, *Environment and Planning* **9A**: 285–344.

Wright, S. J. (2002). Modifying SQP for degenerate problems, *SIAM Journal on Optimization* **13**(2): 470–497.

Wright, S. J. (2003). Constraint identification and algorithm stabilization for degenerate nonlinear programs, *Mathematical Programming* **95**(1): 137–160.