An adaptive trust-region approach for nonlinear stochastic optimization with an application in discrete choice theory

Fabian Bastin

CERFACS, Parallel Algorithms Project
42, Avenue G. Coriolis, 31057 Toulouse Cedex, France
fabian.bastin@cerfacs.fr

Abstract. We consider stochastic nonlinear programs, restricting ourself to differentiable, but possibly non-convex, problems. This leads us to consider non-linear approaches, designed to find second-order critical solutions. We focus here on the use of trust-region approaches when solving a sample average approximation, and adapt the technique to only use sub-samples when possible, adjusting the sample size at each iteration. We finally present an extension to the estimation of mixed logit models, that are popular in discrete choice theory when the population heterogeneity is taken into account.

Keywords. Nonlinear Stochastic Programming, Trust-Region, Monte-Carlo, Discrete Choice, Mixed Logit

1 Nonlinear stochastic programming

A classical problem in stochastic programming is the minimization of the expectation of some function depending on a random variable $\boldsymbol{\xi}$, defined on the probability space $(\Xi, \mathcal{F}, P)$:

$$\min_{z \in S} g(z) = E_{\boldsymbol{\xi}}[G(z, \boldsymbol{\xi})], \tag{1}$$

where S is the feasible set, that we first assume to be deterministic. If the distribution of ξ is continuous or discrete with a large number of possible realizations, $g(z)$ can be very hard to evaluate, and we have to turn to approximations such as Monte Carlo methods (see [1] for a review). The original problem (1) is then replaced by then approximations obtained by generating draws $\xi_1, \ldots, \xi_N$:

$$\min_{z \in S} \hat{g}_N(z) = \frac{1}{N} \sum_{i=1}^{N} G\left(z, \xi_i\right). \tag{2}$$

We refer to (1) and (2) as the true (or expected value) and the sample average approximation (SAA) problems, respectively. If S is defined as

$$S = \left\{ x \in V \subset \mathcal{R}^m \text{ such that } \begin{cases} c_j(z) = E_P[H_j(z, \boldsymbol{\omega})] \geq 0, \ j = 1, \ldots, k, \\ c_j(z) = E_P[H_j(z, \boldsymbol{\omega})] = 0, \ j = k+1, \ldots, C. \end{cases} \right\},$$

where V is assumed to be compact, we can also define SAAs of c_j $(j = 1, \ldots, C)$.

If the random vectors $\boldsymbol{\xi}_k$, $k = 1, \ldots, \infty$, are assumed to be independent and identically distributed (IID), from the Kolmogorov consistency theorem, we can construct the infinite-dimensional probability space $(\Xi_\Pi, \mathcal{F}_\Pi, P_\Pi)$, whose elements are processes of the form $\overline{\xi} = \{\xi_k\}_{k=1}^{\infty}$. We will denote by $\hat{z}_N^*(\overline{\xi})$ a solution of the SAA problem (2). Since V is compact: $z_N^*(\overline{\xi})$ has some limit point $z^*\left(\overline{\xi}\right)$ as $N \to \infty$. We therefore assume (wlog) that $z_N^*(\overline{\xi})$ converges to $z^*\left(\overline{\xi}\right)$ as $N \to \infty$, by considering a

Dagstuhl Seminar Proceedings 05031
Algorithms for Optimization with Incomplete Information
http://drops.dagstuhl.de/opus/volltexte/2005/55

subsequence if necessary. If the approximate solutions $z_N^*(\overline{\xi})$ are first-order critical for the corresponding problems, under some regular conditions, $z^*\left(\overline{\xi}\right)$ satisfies first-order conditions for (1) [1]. Requiring that $z_N^*(\overline{\xi})$ are second-order critical does not however ensure that $z^*\left(\overline{\xi}\right)$ is also second-order critical for (1) [2]. Consistency results can nevertheless be obtained by enforcing stronger assumptions. We first define $\epsilon_N\left(z,\overline{\xi}\right)$ as

$$\epsilon_N(z,\overline{\xi}) = \begin{pmatrix} \hat{g}_N(z) - g(z) \\ \hat{c}_{jN}(z) - c_j(z),\ j=1,\ldots,M \\ \nabla_z \hat{g}_N(z) - \nabla_z g(z) \\ \nabla_z \hat{c}_{jN}(z) - \nabla_z c_j(z),\ j=1,\ldots,M \end{pmatrix}.$$

We then have the following property.

Assume that, for almost every $\overline{\xi}$ in $(\Xi_\Pi, \mathcal{F}_\Pi, P_\Pi)$, $\lambda^*\left(\overline{\xi}\right)$ is the unique vector of Lagrangian multipliers associated to the original program at $z^*\left(\overline{\xi}\right)$, that

(a) $\epsilon_N(z_N^*(\overline{\xi}),\overline{\xi}) \longrightarrow 0$ uniformly on V, as $N \to \infty$,
(b) $z_N^*(\overline{\xi}) \longrightarrow z^*\left(\overline{\xi}\right)$ as $N \to \infty$, $z^*(\overline{\xi})$, $z_N^*(\overline{\xi})$ do not belong to the boundary of V,
(c) $\nabla_{zz}^2 \hat{g}(z_N^*(\overline{\xi}), \epsilon_N(z_N^*(\overline{\xi}),\overline{\xi})) \longrightarrow \nabla_{zz}^2 g\left(z^*\left(\overline{\xi}\right)\right)$ as $N \to \infty$,
(d) $\nabla_{zz}^2 \hat{c}_j(z_N^*(\overline{\xi}), \epsilon_N(z_N^*(\overline{\xi}),\overline{\xi})) \longrightarrow \nabla_{zz}^2 c_j\left(z^*\left(\overline{\xi}\right)\right)$ $(j=1,\ldots,M)$ as $N \to \infty$.

and that the strict complementarity condition holds at $\left(z^*\left(\overline{\xi}\right), \lambda^*\left(\overline{\xi}\right)\right)$. Assume furthermore that the Jacobian of the KKT conditions is almost surely nonsingular at $\left(z^*\left(\overline{\boldsymbol{\xi}}\right), \lambda^*\left(\overline{\boldsymbol{\xi}}\right)\right)$. Then $(z^*(\overline{\boldsymbol{\xi}}), \lambda^*(\overline{\boldsymbol{\xi}}))$ almost surely satisfies the second-order sufficient conditions for the original program.

Proof and other results can be found in [2].

2 An adaptive trust-region algorithm

The SAA problem can be solved using a trust-region approach. We consider here the unconstrained case. The main idea is to compute at iteration k (with current estimate z_k) a trial point $z_k + s_k$ by approximately minimizing a model of the objective function, for instance the quadratic model

$$m_k(z_k+s) = \hat{g}_N\left(z_k\right) + \left\langle \nabla_z \hat{g}_N\left(z_k\right), s\right\rangle + \frac{1}{2}\langle s, H_k s\rangle, \tag{3}$$

where H_k is a symmetric approximation of the Hessian $\nabla_{zz}^2 \hat{g}_N(z_k)$, inside a trust region $\mathcal{B}_k$ defined as $\{z \in \mathcal{R}^m \,|\, \|z - z_k\| \leq \Delta_k\}$, where Δ_k is called the trust-region radius. The predicted and actual decreases are then compared by computing the ratio

$$\rho_k = \frac{\hat{g}_N(z_k+s_k) - \hat{g}_N(z_k)}{m_k(z_k+s_k) - m_k(z_k)}.$$

If ρ_k is greater than a certain threshold η (for instance 0.01), the trial point becomes the new iterate, and the trust-region radius is (possibly) enlarged. The trial point is otherwise rejected and the trust region is shrunk, in order to improve the correspondence of the model with the objective function [3].

The trust-region approach can be easily adapted to include a variable sample size strategy [4], based on the idea of generating a full set of draws prior to optimization, but to only use part of it during certain stages of the optimization process. At a given iteration k, using N_k draws, we compute a candidate sample size N^+ in $[N_{\min}^k, N_{\max}]$, where $N_{\max}$ is the final sample size, and $N_{\min}^k$ is the minimum number of draws. If the ratio τ_k between the decrease in the model and the estimated

accuracy is greater than 1, we set N^+ to the minimum of $\lceil 0.5N^{\max} \rceil$ and the size needed to obtain an accuracy equal to the model decrease, denoted by N^s. If the improvement is smaller than the precision, but greater than the ratio between the sample size N^s and N^+, we set R^+ to the minimum of $\lceil 0.5N_{\max} \rceil$ and $\lceil \tau_k N^s \rceil$, on the grounds that an increase of the order of the estimated accuracy could be reached in approximately $\lceil \tau_k \rceil$ iterations. Otherwise, we set N^+ to $\lceil 0.5N_{\max} \rceil$ as long as τ_k is greater than some threshold, and to $N_{\max}$ when this condition is not met. We then compute the approximate objective function with N^+ draws at the trial iterate. If the ratio ρ_k is less than a constant η, we recompute the approximate objective at z_k with N^+ draws if $N_k < N^+$, in order to take account of variance difference, or, when $N_k > N^+$, we set $N^+ = N^k$. We then again compute the ratio ρ_k, with updated sample sizes. As a safeguard, we also increase the minimum sample size if the algorithm exhibits poor performance due to the variations of accuracy. The algorithm stops when the gradient norm is less than a pre-defined tolerance, or a fraction of the estimated accuracy, where we expect that no more significant decrease in the objective will be achieved. Convergence results can be found in [4].

3 Application to Mixed Logit models estimation

Discrete choice modeling is concerned with the description of choice behavior among a finite set of alternatives. An individual i $(i = 1, \ldots, I)$ is assumed to select in the set $\mathcal{A}(i)$ the alternative that maximizes his/her utility, expressed as $U_{ij}(\beta_j, x_{ij}) = V_{ij}(\beta_j, x_{ij}) + \boldsymbol{\epsilon}_{ij}$, where β_j is the vector of model parameters, and x_{ij} are the observed attributes of alternative j, while $\boldsymbol{\epsilon}_{ij}$ is a random term reflecting the unobserved part. If the terms $\boldsymbol{\epsilon}_{ij}$ are independently Gumbel distributed, we obtain the classical multinomial logit model, characterized by the logit formula expressing the probability that the individual i chooses alternative j:

$$L_{ij}(\beta) = \frac{e^{V_{ij}(\beta)}}{\sum_{l=1}^{|\mathcal{A}(i)|} e^{V_{il}(\beta)}}.$$

Mixed logit models relax the assumption that the explanatory variables β are the same for all individuals, by assuming instead that individual explanatory variables vectors $\beta(i)$ $(i = 1, \ldots, I)$, are realizations of a random vector $\boldsymbol{\beta}$, derived from a random vector $\boldsymbol{\gamma}$ and a parameters z. The probability choice is then $P_{ij}(z) = E_P\left[L_{ij}\left(\boldsymbol{\gamma}, z\right)\right]$, and z is estimated by maximizing the log-likelihood function:

$$\max_z LL(z) = \max_z \frac{1}{I} \sum_{i=1}^{I} \ln P_{ij_i}(z), \tag{4}$$

where j_i is the (observed) alternative choice made by the individual i. The corresponding SAA problem is

$$\max_z SLL^N(z) = \max_z \frac{1}{I} \sum_{i=1}^{I} \ln \frac{1}{N} \sum_{n_i=1}^{N} L_{ij_i}(\gamma_{n_i}, z). \tag{5}$$

The analogy with stochastic programming can also be used to derive consistency results for a fixed population size [2]. Moreover, $SLL^N(z)$ can be shown to be an asymptotically unbiased estimator of $LL(z)$, with an estimable asymptotic value of the confidence interval radius [2,5], that we use as the accuracy evaluation. The simulation bias (resulting from the logarithm operator) can also be approximated and the BTRDA algorithm can be adapted in order to take it into account, in addition to the Monte-Carlo accuracy. As an illustration, we consider the estimation of a parking

choice model, described in [6], with 9 parameters (1 constant, 5 normally distributed, 3 log-normally distributed), 1335 observations, 298 individuals. Each individual delivers several observations, but heterogeneity is only considered at the population level, so the approximate probability choice is $SP_{ij_i}^N = \frac{1}{N}\sum_{n=1}^{N}\prod_{t=1}^{T_i} L_{ij_i}^t(\gamma_n, z)$. The evolution of the number of draws with the iteration index and log-likelihood value is illustrated in Figure 1.

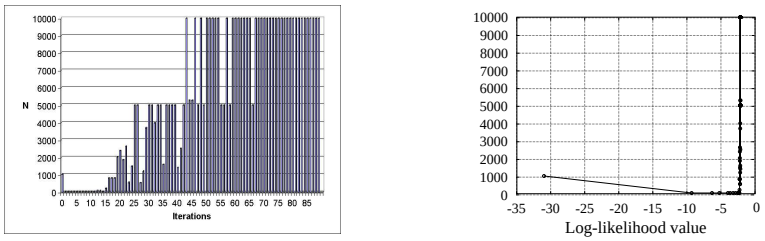

Fig. 1. number of draws evolution

4 Conclusion and future research

Nonlinear stochastic programming opens various issues, both theoretical and practical. Only local minimizers can usually be guaranteed; this has pushed us to to revisit consistency; in particular, while first-order consistency can be ensured under regular conditions, second-order consistency requires more care. First results have been proposed, but more research is needed to relax some of the assumptions and better assess the quality of the approximate solutions. On the numerical point of view, classical algorithms have to be adapted in order to take account of the structure, and to increase the efficiency. We have here introduced a trust-region algorithm for SAA problems, with an extension in mixed-logit models estimation. This algorithm allows to vary the number of used draws from iteration to iteration. The method could however be refined, for instance by combining external and internal sampling strategies, and extended to other class of problems and approximations techniques, as soon as the error can be evaluated.

References

[1] Shapiro, A.: Monte Carlo sampling methods. In Shapiro, A., Ruszczyński, A., eds.: Stochastic Programming. Volume 10 of Handbooks in Operations Research and Management Science. Elsevier (2003) 353–425
[2] Bastin, F., Cirillo, C., Toint, Ph.L.: Convergence theory for nonconvex stochastic programming with an application to mixed logit. Mathematical Programming, Series B (Submitted)
[3] Conn, A.R., Gould, N.I.M., Toint, Ph.L.: Trust-Region Methods. SIAM, Philadelphia, USA (2000)
[4] Bastin, F., Cirillo, C., Toint, Ph.L.: An adaptive monte carlo algorithm for computing mixed logit estimators. Computational Management Science (Submitted)
[5] Gouriéroux, C., Monfort, A.: Simulation-Based Econometric Methods. Oxford University Press (1996)
[6] Hess, S., Polak, J.: Mixed logit estimation of parking type choice. Presented at the 83rd Transportation Research Board Annual Meeting (2004)