Details of estimation

Elaborating on "Time and Causality..."

Simen Gaure, Knut Røed, Tao Zhang

The Frisch Centre for Economic Research

12/1-2007

1 Maximization strategy

We have the following log-likelihood:

$$\mathcal{L}(\beta, W, v, p) = \sum_{i=1}^{N} \log L(X_i, \beta, W, v, p)$$

where L is the likelihood-contribution from a single individual. N is the number of individuals, X_i is a vector of covariates, β is a vector of parameters, W is the number of support points in our heterogeneity distribution, p is a W-dimensional vector, the probabilities in the heterogeneity distribution, v is a $K \times W$ matrix, the location points of the heterogeneity distribution. K is the number of transitions, in this application $K = 2$.

Note that to avoid constrained maximization we have parametrized the probabilites $p = (p_1, \ldots, p_W)$ as

$$p_j = \frac{\exp(\xi_j)}{\sum_k \exp(\xi_k)}$$

with $\xi_1 = 0$ and $\xi_j \in \mathbb{R}$ for $j > 1$. Whenever we say we optimize with respect to p, we mean to optimize w.r.t $(\xi_j)_{j=2}^{W}$.

Our task is to maximize the function $\mathcal{L}(\beta, W, v, p)$ with respect to (β, W, v, p), subject to the constraint $W \leq N$ (this is from [4]). Fortunately, [3] has taken care of some of it. We may handle W in the following way.

1. Set $W = 1$ so that $p = 1$, draw random β and v.
2. Do a maximization of $\mathcal{L}(\beta, W, v, p)$ w.r.t (β, v, p), using (β, v, p) from the previous step as an initial value. If the new likelihood is an *improvement* go to step 3, otherwise terminate.
3. Increase W by 1, and search for a *better* likelihood $\mathcal{L}(\beta, W, v, p)$ by varying (v, p) while keeping β fixed. Go to step 2.

[1] The authors wish to thank Angelo Melino for fruitful discussions.

Some words are emphasized above, in step 3, we need to specify what we mean by *better*. In step 2 we need to specify what we mean by *improvement*. We also need to decide how to search in step 3, and how to maximize in step 2. When we have finished, we need to decide which model (i.e. which W) to use.

Note that the algorithm given here is slightly different from both the one in [1] and the one following [3, Theorem 3.5]. In step 3, we could terminate the algorithm if we can't improve the likelihood (in the sense that a certain directional derivative can't be made positive), but this is conditional on the fixed β. There is a chance that step 2 may actually find a better likelihood (by changing β slightly), even if step 3 can't do it.

1.1 The search for a new masspoint

In step 3 we need to search for a new masspoint. That is, we should find a new location vector and a probability which increases the likelihood. Following [3, Theorem 3.5] we do this as follows:

Let $p = (p_1, \ldots, p_W)$ and $v = (v_{km})$ be the old probabilities and location points. Let w be a K-dimensional vector. Let

$$\mathcal{M}(w, \rho) = \mathcal{L}(\beta, W + 1, [vw], ((1 - \rho)p, \rho))$$

where $[vw]$ is the matrix v extended (by concatenation) with the column w. I.e. $\mathcal{M}(w, \rho)$ is the log-likelihood with $W + 1$ points, the new location vector is w and the new probability is ρ.

We have tried out a couple of different implementations of step 3.

1.1.1 Heckman and Singer's suggestion

Form the derivative in $\rho = 0$ as in [3, p. 304]:

$$G(w) = \frac{\partial \mathcal{M}}{\partial \rho}(w, 0) = \lim_{\rho \to 0} \frac{\mathcal{M}(w, \rho) - \mathcal{M}(w, 0)}{\rho}.$$

We search for a K-dimensional vector w which makes $G(w)$ positive.

There are several ways to do this, e.g. local maximization, grid search, adaptive grid search, etc. We have picked *simulated annealing* as implemented by [2] (to be found at http://www.netlib.org/opt), but modified it to terminate when we reach a positive function value. For our purposes we have usually limited each coordinate to the interval $[-5, 1]$. We have used somewhat varying annealing parameters, among them the set $N_S = 5$, $N_T = 5, N_\epsilon = 3$, initial temperature 1.0, a temperature reduction of 0.4 and max evaluation limit $10K^2N_SN_T$=1000.

For the case $K = 2$ we could pull off a more comprehensive K-dimensional grid search, but as we have seen this is not necessary. We have developed the code for an arbitrary number of transitions and we have successfully estimated datasets with $K = 7$, then a grid search would take more time than we have.

Unlike [3] and [1], we do not attempt to maximize $G(w)$, we stop as soon as $G(w)$ becomes positive. Once we have found such a w we use it as the new location vector $v_{\cdot W+1}$, we set $\rho = 10^{-5}$ and use $((1-\rho)p, \rho)$ as the new probabilities.

The reason we don't keep ρ at 0 in this initial vector is that formally we can't represent a zero probability with the aforementioned probability parametrisation. Setting the corresponding ξ_{W+1} to a large negative number will cause ρ to be numerically equal to 0, but we will not be able to get any good gradients. This is a consequence of the way we parametrise the probabilities.

Maximizing $G(w)$ sometimes performs worse (in terms of cpu-time), and typically does not perform better than terminating on positive $G(w)$. The reason it performs worse is partly that maximization takes time, and partly that it often forces w to the boundary of its domain which often is a numerically less favourable initial value for step 2. The same phenomenon was reported in [1, Section 6.3]. As an alternative, they maximize an *approximation* to $\mathcal{M}(w, \rho)$, we have not tried their method.

All we require from step 3 is an initial vector w for step 2, with $G(w) > 0$ so that $M(w, \rho) \geq M(w, 0)$ in a neighbourhood around $\rho = 0$. We are not aware of any foundation for claiming that a w which maximizes $G(w)$ is a particularly good choice. We have no reason to believe that spending a lot of time on maximizing G beyond 0 in step 3, subsequently will yield a better likelihood in step 2.

There is a drawback with not maximizing $G(w)$. Since there is then a random component in the search, our results will not be exactly reproducible. That is, estimating the same dataset several times may lead to different results. Even if the algorithm terminates with the same heterogeneity distributions and parameter estimates, the points may have been found in different order. If we use AIC as selection criterion, the final results may be different. However, we have yet to see any dramatic effects of this.

1.1.2 A more direct approach

Since differentiation is an unbounded linear operator, the derivative of a function may easily be more erratic than the function itself. This may create difficulties in our search. We have also tried a more direct way. We fix ϵ e.g. at 10^{-4}, and search for a w which maximizes $M(w, \epsilon)$, or at least search for a w with $M(w, \epsilon) > M(w, 0)$.

This method is more robust than maximizing the directional derivative, most of the time it performs as well as the search for a positive derivative described above. It does typically not end up on the boundary. It is slightly worse for finding the final few points of support (typically those with probability less than ϵ and a minimal improvement in likelihood). When using AIC for picking the best model, the difference is neglible. Also, the fallback method in section 1.1.3 alleviates this problem.

1.1.3 When step 3 fails

Sooner or later, step 3 will fail to find a better likelihood, this happens with certainty when there are no more points to add. If we were certain that we have

found the correct β's, and that our global search was perfect, we could then terminate the algorithm. Alas, our world is not perfect.

It may happen that step 3 fails to find a better likelihood for more prosaic reasons, like numerical inaccuracies, or lack of luck in a slightly stochastic search. (Our simulated annealing may anneal too fast.)

Recognizing that our global search is not perfect, we have a final simplistic fallback in this case.

We draw a new distribution (v, p) at random and continue with step 2.

Alternatively, we use the best w we have, i.e. we have done a global maximization and use the w which maximizes $G(w)$ or $M(w, \epsilon)$.

Both methods sometimes yield a better likelihood in step 2, this is most often the final masspoint, with a low probability and a very small likelihood improvement.

If using AIC as selection criterion, this step is, for our data, unnecessary, i.e. we can then do as suggested by [3] and terminate the algorithm if step 3 fails.

1.2 The maximization

The maximization in step 2, being likelihood maximization, should really be a global maximization. We have some real world datasets with $N \approx 10^6$, $K = 7$, $M \approx 50$ and $\beta \in \mathbb{R}^{3000}$, global maximization is then beyond our current computational capabilities. We have resorted to local maximization, more specifically a combination of BFGS and Fisher scoring (i.e. Newton's method with the Hessian replaced by the Fisher matrix).

Moreover, we have divided the maximization into two parts. We have seen that often the maximization leads to quite small changes in β, so we first maximize with respect to (v, p), then w.r.t. (β, v, p). For high-dimensional β, this is very favourable, in particular for the BFGS algorithm whose typical convergence time is proportional to the number of variables.

We also insert a check for degenerate heterogeneity distribution here.

So our previous step 2 looks like this:

2.1 Keep β fixed, maximize $\mathcal{L}(\beta, W, v, p)$ w.r.t. (v, p).

2.2 If one of the probabilities in p is zero, remove the point. If two of the columns of v are equal, say c_i and c_j, set c_i to $0.5(c_i + c_j)$ and add p_j to p_i. Remove point j.

2.3 Maximize w.r.t. (β, v, p). Continue with step 2.2 with the modification that if no removal is done, we go on to step 3.

Note that none of the removal operations in 2.2 changes the log-likelihood $\mathcal{L}$.

A probability of less than 10^{-5} is considered to be zero. Two location vectors $v_{\cdot i}$ and $v_{\cdot j}$ are considered to be equal if $\|v_{\cdot i} - v_{\cdot j}\|_\infty < 0.05$. For the datasets in question we have not had any problems with equal location vectors.

By *maximize* above, we mean the following combination of BFGS and Fisher scoring:

a. Set b to 50. Set n to 30.

b. Do up to b iterations of BFGS.

c. Do up to n Newton iterations. If convergence is achieved, terminate the algorithm with success. If $n < 60$ then increase b by 100, n by 10 and go to step b. If $n \geq 60$, terminate with failure.

It turns out that both BFGS and Newton may run into problems, but the above combination often gets around problematic regions. The number of iterations and the convergence criterion may be adjusted.

We have used the L-BFGS-B implementation of BFGS [6] essentially unchanged.

We have used the TNPACK package [5] for the Newton iterations. This is a truncated Newton method with a conjugate gradient step for sparse systems. We can actually afford to solve the Newton equations directly, so we have modified TNPACK to try this first.

2 Complications

The maximization algorithm may sometimes estimate a very large negative value for a heterogeneity parameter. E.g. v_{26} is very negative. Then $e^{v_{26}}$ is *numerically* equal to zero and we are not able to compute meaningful gradients, nor refine v_{26}. The Fisher matrix becomes degenerate, and we can't invert it to obtain standard errors for any of the parameters. In this case we *mark* v_{26} as $-\infty$ and leave it as a constant in the estimation.

Our complications do far from end here. In particular for discrete durations we have occasional numerical problems. I.e. we have expressions like $1 - e^{-e^x}$ and their derivatives. For certain arguments we then use a Taylor approximation to alleviate numerical difficulties.

Also, for some expressions it's more accurate to work directly with their logarithms.

There's also a trick which has proved to be very useful. During estimation, we sometimes come across parameter values which make the likelihood of some individuals numerically equal to zero. That is, for some i we have $L(X_i, \beta, W, v, p) \approx 0$. The gradient for these individuals may become erratically large, and most maximization algorithms will terminate with an error status. This may happen even if the *maximizing* parameter values have no such undesirable properties. It is a consequence of a poor choice of initial values, or unfortunate path-finding by the maximization algorithm.

We have solved this problem by temporarily setting the log-likelihood L for these individuals to $\log(\epsilon)$ where ϵ is the smallest positive real which can be represented in the machine, and we set their gradient to 0. This introduces a discontinuity in the gradient, but it turns out that the maximization algorithm most often survives this and moves away from such regions by relying on the gradient contribution from the "sane" individuals.

2.1 Time

Some computations are time-consuming. The above algorithm has been written in the de facto standard for high performance computing, FORTRAN. We have used MPI to parallelize the program (over the dataset). Most computations have been done on from 8 to 100 cpus of the Titan-cluster operated by the University of Oslo.

References

[1] M. Baker and A. Melino, *Duration Dependence and Nonparametric Heterogeneity*, J. Econometrics **96** (2000), 357–393.

[2] W. L. Goffe, G. D. Ferrier, and J. Rogers, *Global Optimization of Statistical Functions with Simulated Annealing*, J. Econometrics **60** (1994), 65–99.

[3] J. Heckman and B. Singer, *A Method for Minimizing the Impact of Distributional Assumptions in Econometric Models for Duration Data*, Econometrica **52** (1984), 271–320.

[4] B. G. Lindsay, *The Geometry of Mixture Likelihoods: A General Theory*, Ann. Stat. **11** (1983), 86–94.

[5] T. Schlick and Aa. Fogelson, *Algorithm 702: TNPACK — a truncated Newton minimization package for large-scale problems: I. Algorithm and usage*, ACM Trans. Math. Soft. **18** (1992), no. 1, 46–70.

[6] C. Zhu, R. H. Byrd, and J. Nocedal, *Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization*, ACM Trans. Math. Soft. **23** (1997), no. 4, 550–560.