Derivative Free Gradient Projection Algorithms for Rotation

August 5, 2003

A simple modification substantially simplifies the use of the gradient projection (GP) rotation algorithms of Jennrich (2001, 2002). These algorithms require subroutines to compute the value and gradient of any specific rotation criterion of interest. The gradient can be difficult to derive and program. It is shown that using numerical gradients gives almost precisely the same results as using exact gradients. The resulting algorithm is very easy to use because the only problem specific code required is that needed to define the rotation criterion. The computing time is increased when using numerical gradients, but it is still very modest for most purposes. While used extensively elsewhere, numerical derivatives seem to be under utilized in statistics.

1 Introduction

This note introduces a simple very useful modification to the gradient projection (GP) algorithms of Jennrich (2001, 2002). These algorithms are applicable to orthogonal and oblique rotation, and are themselves simple and very general, applying not just to factor analysis, but to other forms of rotation as well. As there name suggests GP algorithms use gradients of the rotation criterion they are designed to optimize. For many standard methods of rotation these gradients are easy to derive and implement, but this need not be the case, especially when experimenting with new methods or comparing a variety of methods. We will show that the gradients in GP algorithms may be replaced by numerical approximations with essentially no effect on the results produced. The numerical gradients may be produced using a simple general method that is not problem specific. Thus the only thing required for a specific application is a definition of the criterion used.

Browne (2001) has given a rotation algorithm using pairwise rotation and line searching that is very general and like the derivative free GP algorithm requires only a definition of the criterion used. Its use is restricted to factor analysis applications or at least to applications where the argument of the rotation criterion is square. Since most applications of rotation are to factor analysis this is a minor restriction.

The use of numerical gradients generally requires more computer time than using exact gradients, but even on large problems this is small. For example for a quartimin rotation with 100 variables and 10 factors the com-

puter time using numerical gradients was 12.97 seconds compared to .157 seconds using exact gradients. For research purposes one may well be willing to trade 12.97 seconds of computer time for the personal effort required to derive gradient formulas and implement them. When producing software for a specific form of rotation that will be extensively used, however, exact derivatives are probably the appropriate choice.

The basic GP algorithms used here employ a minor modification to those of Jennrich (2001, 2002). This is discussed in Section 3. Matlab (1995) code employing this modification with and without the numerical gradient modification may be downloaded from `http://www.stat.ucla.edu/research/gpa`.

2 The rotation problem

Let $\mathcal{R}$ denote the set of all k by m matrices with $k \geq m$ and let f be a function defined on $\mathcal{R}$. The general orthogonal rotation problem is to minimize

$$f(T) \quad \text{given} \quad T \in \mathcal{O}$$

where $\mathcal{O}$ is the set of all T in $\mathcal{R}$ with orthonormal columns. The general oblique rotation problem is to minimize

$$f(T) \quad \text{given} \quad T \in \mathcal{N}$$

where $\mathcal{N}$ is the set of all T in $\mathcal{R}$ with normal columns, that is columns of length one.

For rotation in factor analysis the matrices in $\mathcal{R}$ are square and f has a

special form. For orthogonal rotation

$$f(T) = Q(AT)$$

and for oblique rotation

$$f(T) = Q(A(T')^{-1})$$

where Q is a factor analysis rotation criterion, for example quartimin, and A is an initial loading matrix.

3 Basic GP algorithms

Jennrich (2001, 2002) has given GP algorithms for orthogonal and oblique rotation. To simplify presentation consider the general problem of minimizing

$$f(T) \quad \text{given} \quad T \in \mathcal{M}$$

where $\mathcal{M}$ is an arbitrary submanifold of $\mathcal{R}$. When $\mathcal{M} = \mathcal{O}$ this is the general orthogonal rotation problem and when $\mathcal{M} = \mathcal{N}$ it is the general oblique rotation problem.

What makes the orthogonal and oblique GP algorithms work is that in in either case it is easy to project an arbitrary X in $\mathcal{R}$ onto $\mathcal{M}$. Let $\rho(X)$ denote the projection. The basic GP algorithm proceeds as follows. Let T be in $\mathcal{M}$ and $G = df/dT$ be the gradient of f at T. A step in the GP algorithm updates T to

$$\tilde{T} = \rho(T - \alpha G) \tag{1}$$

Jennrich (2001,2002) has shown that if T is not a stationary point of f restricted to $\mathcal{M}$, replacing T by $\tilde{T}$ will decrease $f(T)$ for all sufficiently small $\alpha > 0$. Using this, partial stepping produces a strictly monotone algorithm for minimizing f restricted to $\mathcal{M}$.

Here we use a modification of this procedure motivated by a desire to simplify and improve the partial stepping procedure. Let G_p be the projection of G onto the linear manifold tangent to $\mathcal{M}$ at T. Jennrich (2001,2002) has shown that T is a stationary point of f restricted to $\mathcal{M}$ if and only if $G_p = 0$. The GP algorithm is stopped when G_p is close to zero. This is a useful stopping rule because when it stops does not depend on the speed of the algorithm or even on the algorithm used. The modified GP algorithm used here replaces G in (1) by G_p. This is a minor modification because G_p is already computed for use in the stopping rule. The update with this replacement is

$$\tilde{T} = \rho(T - \alpha G_p) \tag{2}$$

Like the update in (1) it has the property that if T is not a stationary point of f restricted to $\mathcal{M}$, replacing T by $\tilde{T}$ decreases $f(T)$ whenever $\alpha > 0$ is sufficiently small. Orthogonal and oblique GP algorithms with this modification may be downloaded from the web site given above.

4 Derivative free GP algorithms

GP algorithms use the gradient $G = df/dT$ of f at points T in $\mathcal{M}$. The derivative free version of the GP algorithm approximates G by using nu-

merical derivatives. There are several methods of approximation that might be used. One may, for example, use the forward differences found in a first course in calculus, symmetric differences, or some form of Richardson extrapolation (see e.g. Conte and deBoor, 1980). Because they are simple, are frequently used, and work quite well in our applications, we have used symmetric differences. More specifically, we approximated the partial derivative of $f(T)$ with respect to the component t_{rs} of T using

$$G_{rs} = \frac{\partial f}{\partial t_{rs}} \approx \frac{f(T + \delta J(r,s)) - f(T - \delta J(r,s))}{2\delta}$$

where $J(r,s)$ is a k by m matrix with a one in the (r,s) position and zeros elsewhere and δ is a small increment. Note that G is approximated one component at a time. While choosing an appropriate δ is a potential problem, the value $\delta = .0001$ has never failed in our examples.

5 Testing the numerical derivatives

To test the efficacy of using numerical derivatives we will compare results of using numerical and exact derivatives in the context of oblique quartimin (Carroll, 1953) and simplimax (Kiers, 1994) rotation in factor analysis. Factor analysis was chosen because it represents the most common area of application of rotation. Oblique rotation was chosen because it provides a larger class of potential rotations and is generally more difficult than orthogonal rotation. Quartimin was chosen because it is extensively used and tends to be insensitive to starting values. Finally simplimax was chosen because it

is a promising newer method that differs considerably from quartimin. We will evaluate the precision of the results produced using numerical derivatives and the computer times required.

5.1 Numerical precision

Let Λ_e be the rotated loading matrix produced by a GP algorithm using exact derivatives and Λ_n be the corresponding loading matrix produced using the numerical derivatives described above. We will compute the minimum number of decimal places of agreement between the components of Λ_e and Λ_n. More specifically we will compute

$$\text{agre} = -\log_{10} \max_{rs} |\lambda_{rs}^{(n)} - \lambda_{rs}^{(e)}|$$

where $\lambda_{rs}^{(n)}$ and $\lambda_{rs}^{(e)}$ denote the (r, s) components of Λ_n and Λ_e respectively.

To generate data for the quartimin comparisons let

$$A_0 = I \otimes u$$

where I is a k by k identity matrix, u is a column vector of b ones, and $\otimes$ denotes the Kronecker product. Then A_0 has perfect simple structure, that is it has at most one nonzero element in each row. Let Z be a $p = bk$ by k matrix whose components are independent standard normal variables and let

$$A = A_0 + .25Z \tag{3}$$

Using 50 variables and 5 factors, that is $b = 10$ and $k = 5$, 100 independent realizations of A were generated for use as initial loading matrices. Note that

these A are not particularly similar nor are their rotations. This is because the random term $.25Z$ is not particularly small.

The quartimin GP algorithm using exact and numerical derivatives was applied to each initial loading matrix and the minimum number of digits of agreement computed. All rotations used an identity start. The algorithms converged to a stationary point in every case and required the same number of iterations in all but one case where the numerical derivative algorithm required one less iteration than the exact derivative algorithm. The number of iterations required ranged from 23 to 76. The minimum number of digits of agreement ranged from 6.86 to 8.64 with a median of 8.29. For many applications this degree of agreement is almost perfect. This agreement was for the rotations produced directly. No alignment, that is column sign change or permutation, was required.

For the simplimax comparisons Thurstone's (1947) well known 26 variable box data was used. This may also be found in Kiers (1994). The simplimax criterion is not continuous, but Kiers has shown how it may be optimized using an iteratively re-weighted pairwise rotation algorithm. Jennrich (2002) has shown this may also be done by replacing cycles of pairwise rotations by GP steps. We will compare numerical and exact gradient versions of this GP simplimax algorithm. The simplimax criterion tends to have local minima. To deal with this Kiers recommends using the best of a number of random starts. We will compare results obtained from 10 random starts, the number used by Kiers for the box data. By a random start we mean a rotation matrix T whose columns are statistically independent and uniformly

distributed over the unit sphere in k dimensions. The exact and approximate gradient algorithms converged to a stationary point from each of the 10 random starts and required the same number of iterations from all but one start where the numerical gradient algorithm required one less iteration. The number of iterations required ranged from 18 to 147. The minimum number of digits of agreement ranged from 6.68 to 8.40 with a median of 7.98 which again for many applications is almost perfect. The loading matrix produced by the best of the 10 random starts agreed exactly to the precision displayed with that given by Kiers.

5.2 Time comparison.

When many are required, numerical derivatives can be expensive to compute. To investigate this, independent realizations of initial loading matrices of the form (3) were generated for $b = 10$ and $k = 2, \cdots, 10$. Choosing $b = 10$ is somewhat arbitrary, but using 10 variables per factor is not unreasonable. These realizations were rotated using the quartimin GP algorithm with exact and numerical derivatives. The computing times using a Power Macintosh G3 ranged from .024 to .647 seconds using exact derivatives and from .051 to 21.73 seconds using numerical derivatives. For each $k = 2, \cdots, 10$ the ratio of the time required using numerical derivatives to that required using exact derivatives was computed. These ratios, plotted in Figure 1, ranged from 2.13 for $k = 2$ to 41.83 for $k = 10$. Clearly for a production program, such as a SAS (1999) program, or a simulation study, exact derivatives would be

preferred, but for general research on rotation methods an investigator may be eager to trade seconds of computer time for the effort involved in deriving exact derivatives and implementing the software to compute them.

Exact gradients can be easy to compute. For the quartimin criterion

$$G = -(\Lambda'\Lambda^3 T^{-1})'$$

where Λ^3 is the elementwise cube of Λ (Jennrich, 2002). In other cases, however, exact gradients may be difficult to obtain. McCammon's (1966) entropy criterion has the form

$$Q(\Lambda) = \frac{\sum\sum e\left(\frac{s_{ir}}{s_{.r}}\right)}{\sum e\left(\frac{s_{.r}}{s_{..}}\right)}$$

where $s_{ir} = \lambda_{ir}^2$, $e(x) = -x \log x$, $s_{.r} = \sum_i s_{ir}$, and $s_{..} = \sum_r s_{.r}$. Deriving an exact gradient G is quite difficult requiring, for the author at least, a substantial effort. For other than a large number of applications of McCammon's method, there is a real advantage in using a derivative free approach.

6 Discussion

The main point of this note is very simple. Replace the derivatives in the GP algorithms for rotation by numerical derivatives. This modification is important because it provides an algorithm for optimization of any orthogonal or oblique rotation criterion while requiring the minimum possible problem specific information, namely the definition of the criterion.

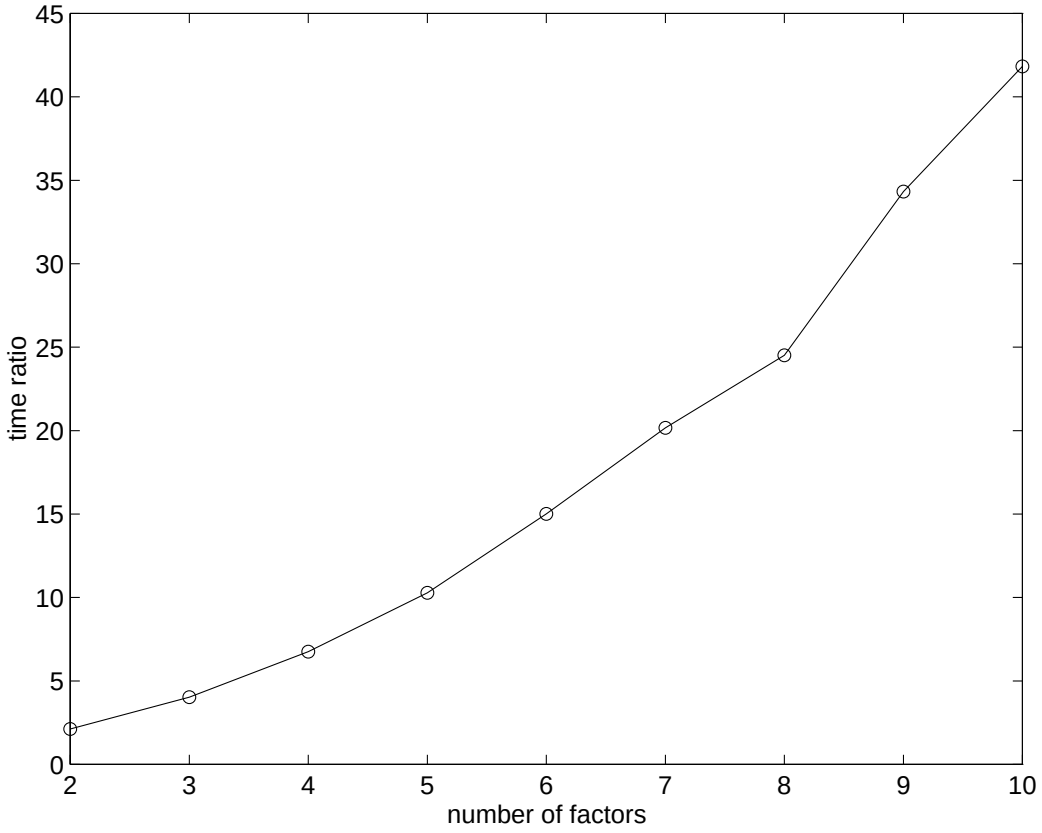

Figure 1: The ratio of the computer time required by the numerical gradient method to that required by the exact gradient method as a function of the number of factors

Although numerical derivatives are extensively used in applied mathematics, they seem to be under-utilized in statistics and psychometrics. A key word search of the Current Index to Statistics for "numerical derivatives" or "numerical differentiation" produced only two references, neither to psychometric journals. This is surprising because as we have seen numerical derivatives can work quite well. That this is often the case seems to be a well kept secret.

7 References

Browne, M.W. (2001). An overview of analytic rotation in exploratory factor analysis. *Multivariate Behavioral Research, 36*, 111-150.

Carroll, J.B. (1953). An analytical solution for approximating simple structure in factor analysis. *Psychometrika, 18*, 23-38.

Conte, S.D. and deBoor, D.B. (1980). *Elementary numerical analysis: An algorithmic approach.* New York: McGraw-Hill.

Jennrich, R.I. (2001). A simple general procedure for orthogonal rotation. *Psychometrika, 66*, 289-306.

Jennrich, R.I. (2002). A simple general method for oblique rotation. *Psychometrika, 67*, 7-19.

Kiers, H.A.L. (1994). Simplimax: Oblique rotation to an optimal target with simple structure. *Psychometrika, 59*, 567-579.

Matlab (1995). The MathWorks Inc., 24 Prime Park Way, Natick, MA, 01760.

McCammon, R.B. (1966). Principal component analysis and its application in large-scale correlation studies. *Journal of Geology, 74*, 721-733.

SAS (1999). *SAS/STAT User's Guide, version 8.* Cary, NC: SAS Institute Inc.

Thurstone, L.L. (1947). *Multiple factor analysis.* Chicago: University of Chicago Press.