THE GRADIENT PROJECTION ALGORITHM FOR ORTHOGONAL ROTATION

1 The problem

Let $\mathcal{M}$ be the manifold of all k by m column-wise orthonormal matrices and let f be a function defined on arbitrary k by m matrices. The general orthogonal rotation problem is to minimize f restricted to $\mathcal{M}$. The most common problem is rotation to simple loadings in factor analysis. There

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

where Q is a rotation criterion, for example varimax, and A is an initial loading matrix. The object is to minimize $f(T)$ over all orthogonal matrices T. In this case $m = k$. A variety of other applications may be found in Jennrich (2001).

2 The gradient projection algorithm

The gradient projection (GP) algorithm to be discussed is essentially the singular value decomposition (SVD) algorithm of Jennrich (2001). It, however, was not recognized that the SVD algorithm was in fact a GP algorithm until a remark to that effect appeared in a paper on oblique gradient projection algorithms (Jennrich, 2002). The SVD algorithm was derived and studied as a majorization algorithm. It now seems simpler and more natural to view

and study it as a GP algorithm and that is the purpose of this note. This approach will also emphasize its similarity to the GP algorithm for oblique rotation.

Let $\|B\|$ denote the Frobenius norm of a matrix B. This is defined by

$$\|B\|^2 = \mathrm{tr}(B'B)$$

With respect to this norm the gradient G of f at T is the matrix of partial derivatives of $f(T)$ with respect to the components of T. Viewing T as a current value, the idea of the GP algorithm is to move in the negative gradient direction an amount $\alpha > 0$, that is from T to $T - \alpha G$. In general $T - \alpha G$ will not be in $\mathcal{M}$. To deal with this it is projected onto $\mathcal{M}$. This is easy to do because the projection of an arbitrary k by m matrix X onto $\mathcal{M}$ is given by

$$\rho(X) = UV'$$

where U and V are matrices from a singular value decomposition $X = UDV'$ of X. This result may be found in Jennrich (2002). It may also be derived easily using target rotation. The simplest form of the GP algorithm is:

Choose $\alpha > 0$ and an initial T in $\mathcal{M}$.

(a) Compute the gradient G of f at T.

(b) Replace T by $\rho(T - \alpha G)$ and go to (a) or stop.

This simple algorithm can and often does work, but the stopping rule is a bit vague and the algorithm's success may depend on the choice of α.

To address the stopping rule problem it is shown in Jennrich (2001) that T is a stationary point of f restricted to $\mathcal{M}$ if and only if

$$G = TS \tag{1}$$

for some symmetric matrix S. It is also shown that this condition may be expressed more explicitly as $s = 0$ where

$$s^2 = \|\text{skm}(T'G)\|^2 + \|(I - TT')G\|^2 \tag{2}$$

Motivated by this we will stop the GP algorithm when $s \leq \epsilon$ for some small $\epsilon > 0$

Let T be a differentiable mapping into $\mathcal{M}$. Then $T'T = I$ and

$$dT'T + T'dT = 0$$

from which it follows that $T'dT$ is skew-symmetric. Let $\text{skm}(B)$ denote the skew-symmetric part of B. To address the choice of α problem we begin by finding the differential of ρ at an arbitrary T in $\mathcal{M}$.

Lemma 1:

$$d\rho_T(dX) = T\text{skm}(T'dX) + (I - TT')dX$$

Proof: Let

$$f(T) = \|X - T\|^2/2$$

Its gradient at T is

$$G = X - T$$

If $T = \rho(X)$, T minimizes and hence is a stationary point of f restricted to $\mathcal{M}$. Using (1)

$$X - T = TS$$

for some symmetric matrix S and hence

$$X = TS$$

for some, but not the same, symmetric matrix S. Differentiating

$$dX = dTS + TdS$$

At $X = T$, $S = I$ and

$$dX = dT + TdS \tag{3}$$

and

$$T'dX = T'dT + dS$$

Using the fact that $T'dT$ is skew-symmetric and dS is symmetric

$$\text{skm}(T'dX) = T'dT$$

Thus

$$T\text{skm}(T'dT) = TT'dT$$

From (3)

$$(I - TT')dX = (I - TT')dT$$

Thus

$$T\text{skm}(T'dX) + (I - TT')dX = dT$$

But $dT = d\rho_T(dT)$ which completes the proof.

We turn next to the basic theorem for choosing α.

Theorem 1: If T is in $\mathcal{M}$, if G is the gradient of f at T, and if

$$h(\alpha) = f(\rho(T - \alpha G))$$

then

$$h'(0) = -\|\mathrm{skm}(T'G)\|^2 - \|(I - TT')G\|^2 \tag{4}$$

Proof: Using Lemma 1,

$$\begin{aligned} h'(0) &= -(G, d\rho_T(G)) \\ &= -(G, T\mathrm{skm}(T'G) + (I - TT'')G) \\ &= -\|\mathrm{skm}(T'G)\|^2 - \|(I = TT')G\|^2 \end{aligned}$$

This completes the proof.

Note that the right hand side of (4) is $-s^2$. Thus if T is not a stationary point, $h'(0)$ is negative and hence

$$f(\rho(T - \alpha G)) < f(T)$$

when α is sufficiently small. We can use this to choose α so the GP algorithm is monotone decreasing. For example consider the algorithm:

Choose $\alpha_0 > 0$ and T in $\mathcal{M}$

(a) Compute s as defined by (2). If $s < \epsilon$, stop.

(b) Compute the gradient G of f at T and set $\alpha = \alpha_0$.

(c) Compute $\tilde{T} = \rho(T - \alpha G)$.

(d) If $f(\tilde{T}) \geq f(T)$ replace α by $\alpha/2$ and go to (c).

(e) If $f(\tilde{T}) < f(T)$ replace T by $\tilde{T}$ and go to (a).

Note that step (a) stops the algorithm when it is sufficiently close to a stationary point. Note that because of Theorem 1, the return to step (c)

from step (d) can only occur a finite number of times. Note also that the algorithm is strictly monotone. That is the value of f at a new value of T is strictly less than the value of f at the previous value of T. While not guaranteed by this alone, strictly monotone algorithms almost invariably converge to stationary points. In the author's experience this has never failed to happen, not even in simulations. Moreover, because stationary points that are not local minima are not points of attraction of a monotone algorithm, strictly monotone algorithms tend to converge stationary points that are local minima.

In our experience $\alpha_0 = 1$ and $\epsilon = 10^{-5}$ are reasonable choices for these parameters. We use an initial $T = I$ except for some simulations where random T were used.

A number of modifications of this algorithm are possible. For example in step (b) one can replace α by 2α rather than setting it equal to α_0. This tends to speed the algorithm when α_o is too large or too small. One can also use Theorem 1 to produce a fancier partial step in step (d). This may be done by approximating $h(\alpha)$ in Theorem 1 by a quadratic that equals $f(T)$ at zero, $f(\tilde{T})$ at α, and has slope $-s^2$ at zero. The minimizer

$$\tilde{\alpha} = \frac{s^2\alpha^2}{2(f(\tilde{T}) - f(T) + s^2\alpha)}$$

of the approximation is an approximate minimizer of h may be used instead of $\alpha/2$ to replace α in step (d). The author has tried, but does use these modifications because the speed of the unmodified algorithm has never been a problem even in simulations.

3 References

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*, in press.