**PROMISING SCALE FACTORS FOR GRADIENT DESCENT**

*Jean-Pierre Amor, School of Business, University of San Diego, 5998 Alcala Park, San Diego, CA 92110, 619-260-2377, jpamor@sandiego.edu*

**ABSTRACT**

It has often been noted that gradient descent methods are sensitive to the scaling of a problem’s data; that is changing the units of measurement of the decision variables alters the direction of descent chosen at each iteration and hence the rate of convergence of the algorithm. In practice, however, the matter of scaling a problem is either ignored or a simple ad hoc rule-of-thumb is used. This paper investigates the question of scaling in a simple prototype setting and derives “optimal” scale factors for accelerating the convergence of gradient descent algorithms.

**INTRODUCTION**

The classical method of steepest descent (SD), also known as gradient descent, for solving

Minimize f(x) (1)

x

where f: is a continuously differentiable function, can be slow to converge and is not invariant under linear transformations of the variables [2] [4] [5]. This method, however, can be generalized by: (1) relaxing the implicit requirement that the norm over be Euclidean and (2) allowing this norm to vary, at each iteration. We call it generalized steepest descent (GSD). When quadratic norms are used, GSD is equivalent to performing SD with a linear transformation of the variables, at each iteration. For certain norms this process can significantly accelerate the convergence of the algorithm; but it is computationally expensive. This research effort, therefore, attempts to rescale the problem variables “optimally,” repeatedly, and at a reasonable cost. We first study the scaling question in a simplified setting in the hope that sensible application of the results in a more realistic environment will be of computational value.

**FORMULATING THE SCALING PROBLEM**

In our simplified setting, for the convex quadratic problem

Minimize ½ x’Qx + c’x (2)

x

where Q is symmetric (sym.) and positive semi-definite (p.s.d.), we seek a change of scales x=Sy, where S is diagonal (diag.) and nonsingular (nonsing.), which results in the fastest convergence of SD to an -optimal solution of

Minimize ½ y’SQSy + c’Sy. (3)

y

The number of iterations required depends in a quite complicated way on Q, c, S, the starting point (xo) and Optimizing with respect to S seems to be out of the question. Some approximation must be used and the most obvious derives from the well-known Kantorovich Ratio (KR) bound on the rate of convergence of SD [3], which, when modified for p.s.d. quadratic forms, says that

; non-optimal (4)

where: *ek* is the error of the approximation *xk* (measured in “payoff” space), is the largest eigenvalue of SQS, and is the smallest nonzero eigenvalue of SQS. The upper bound in (4) is the smallest that can be obtained without using specific information about . Therefore, to accelerate the rate of convergence of SD, we wish to find an S which minimizes this upper bound, i.e.,

Minimize KR(SQS) (5)

S diag. and nonsing.

with the expectation that as the KR bound is improved, the performance of SD will also improve. It can be shown that solving (5) is equivalent to solving

Minimize CN(SQS) (6)

S diag. and nonsing.

where CN(SQS) is the condition number of the matrix SQS. However, since the eigenvalues of SQS depend in a complicated way on the diagonal elements of S, it is not practical to minimize CN(SQS) directly. After number of reformulations, an attractive surrogate for (6) emerges as

Minimize CV(SQS) (7)

S diag. and nonsing.

where CV(SQS) is the coefficient of variation of the eigenvalues of SQS.

**A PROMISING SCALING METHOD**

Let Q in our prototype problem (2), have m ( n) nonzero eigenvalues. Since S is nonsingular, S’QS also has m nonzero eigenvalues. Since we are only interested in an optimal solution of (7), if one exists, we may rewrite it as

Minimize Variance (SQS) subject to: Mean (SQS) =1. (8)

S diag. and nonsing.

Since the nonzero eigenvalues of SQS and those of S2Q are the same, (8) is equivalent to

Minimize Variance (ZQ) subject to: Mean (ZQ) =1. (9)

Z diag. and p.d.

where Z S2. After more reformulations, we can restate (9) in terms of the trace of the matrix ZQ as

Minimize Trace – 1 subject to: Trace (ZQ) = m. (10)

Z diag. and p.d.

Defining z and q as the column vectors consisting of the diagonal elements of Z and Q, respectively, and P as the matrix whose elements are qij2, it follows that any optimal solution of

Minimize z’Pz subject to: q’z = m (11)

z>0

will provide the squares of the desired scale factors.

When rescaling a general convex problem, Q in (2) will represent a local Hessian of the objective function in (1), f(xk). Since we shall not know, a priori, whether Q is p.d. or p.s.d., the approach used for “solving” (11) should not require such knowledge, nor be computationally expensive since scaling will have to be done multiple times. Our approach is to temporarily ignore the positivity conditions, solve the remaining equality constrained problem, and later account for the fact that only a relaxation of (11) has been addressed. Using Lagrange multipliers and invoking the convexity of z’Pz, a necessary and sufficient condition for a solution of the relaxed problem is essentially

Pz = q (12)

Hence our scaling problem is reduced to solving a system of linear equations.

Let z\* denote any solution of (12). If P is p.d., (which is possible even if Q is only p.s.d. [1]), then z\* = P-1q solves this system uniquely. If P is p.s.d., (12) has infinitely many solutions. We may choose any positive values for the independent variables, thereby fixing the remaining variables uniquely, and call this solution z\*. In any event, if z\* > 0, then z\* solves (11) and we have found a set of optimal scale factors for the convex quadratic problem (2). They are

xi = (zi\*)1/2yi = siyi; i = 1,…, n. (13)

Since SD is to be used on the scaled problem (3), this is equivalent to using GSD(Z\*) on the original problem (2). If z\* > 0, we may proceed directly with GSD(Z\*). However, if z\* the direction to be taken from xk is given by dk = -Z\*, which may or may not be a direction of descent. Consequently, it would be wise to test for a negative directional derivative at each xk. (This test is relatively inexpensive since it involves only 2n multiplications.) If dk < 0, we may use Z\* to deflect the (negative) gradient. If not, it might be preferable not to deflect at the current iteration or perhaps to sub-optimize problem (11). One approach for doing this may be to use matrix perturbation methods. Fortunately, based on our preliminary computational experience with quadratic problems, we have found that, in most cases, Z\* > 0. In the general convex case, we may have many opportunities to calculate Z\*’s and it may be best to wait for a Hessian which produces positive zi\*’s.

**REFERENCES**

[1] Bellman, R. *Introduction to Matrix Analysis.* McGraw-Hill, New York, 1970.

[2] Bertsekas, D. *Nonlinear Programming.* Athena Scientific, Belmont, Massachusetts, 1999.

[3] Kantorovich, L. On the method of Steepest Descent. *Math Reviews, 1948, 9,* 308-309.

[4] Luenberger, D. *Introduction to Linear and Nonlinear Programming.* Addison-Wesley, Reading,

Massachussetts, 1973.

[5] Mordecai, A. *Nonlinear Programming: Analysis and Methods.* Dover Publishing, New York, 2003.