September 1994 LIDS-P-2267

A HYBRID INCREMENTAL GRADIENT METHOD[1] FOR LEAST SQUARES PROBLEMS

by

Dimitri P. Bertsekas[2]

Abstract

The LMS method for linear least squares problems differs from the steepest descent method in that it processes data blocks one-by-one, with intermediate adjustment of the parameter vector under optimization. This mode of operation often leads to faster convergence when far from the eventual limit, and to slower (sublinear) convergence when close to the optimal solution. We embed both LMS and steepest descent, as well as other intermediate methods, within a one-parameter class of algorithms, and we propose a hybrid method that combines the faster early convergence rate of LMS with the faster ultimate linear convergence rate of steepest descent.

[1] Research supported by NSF under Grant 9300494-DMI.

[2] Department of Electrical Engineering and Computer Science, M.I.T., Cambridge, Mass., 02139.

1. INTRODUCTION

We consider least squares problems of the form

$$\begin{aligned} &\text{minimize} \quad f(x) = \frac{1}{2}\|g(x)\|^2 = \frac{1}{2}\sum_{i=1}^{m}\|g_i(x)\|^2 \\ &\text{subject to} \quad x \in \Re^n, \end{aligned} \tag{1}$$

where g is a continuously differentiable function with component functions $g_1, \ldots, g_m$, where $g_i : \Re^n \to \Re^{r_i}$. Here we write $\|z\|$ for the usual Euclidean norm of a vector z, that is, $\|z\| = \sqrt{z'z}$, where prime denotes transposition. We also write ∇g_i for the $n \times r_i$ gradient matrix of g_i, and ∇g for the $n \times (r_1 + \cdots + r_m)$ gradient matrix of g. Least squares problems often arise in contexts where the functions g_i correspond to data that we are trying to fit with a model parameterized by x. Motivated by this context, we refer to each component g_i as a *data block*, and we refer to the entire function $g = (g_1, \ldots, g_m)$ as the *data set*.

In problems where there are many data blocks, and particularly in neural network training problems, gradient-like incremental methods are frequently used. In such methods, one does not wait to process the entire data set before updating x; instead, one cycles through the data blocks in sequence and updates the estimate of x after each data block is processed. Such methods include the Widrow-Hoff LMS algorithm [WiH60], [WiS85], for the case where the data blocks are linear, and its extension for nonlinear data blocks. A cycle through the data set of this method starts with a vector x^k and generates x^{k+1} according to

$$x^{k+1} = \psi_m,$$

where ψ_m is obtained at the last step of the recursion

$$\psi_0 = x^k, \qquad \psi_i = \psi_{i-1} - \alpha^k \nabla g_i(\psi_{i-1}) g_i(\psi_{i-1}), \quad i = 1, \ldots, m, \tag{2}$$

and α^k is a positive stepsize. Thus the method has the form

$$x^{k+1} = x^k - \alpha^k \sum_{i=1}^{m} \nabla g_i(\psi_{i-1}) g_i(\psi_{i-1}). \tag{3}$$

We refer to this method, which is just the nonlinear version of the LMS algorithm, as the *incremental gradient method.*

The above method should be contrasted with the steepest descent method, where the data blocks g_i and their gradients are evaluated at the same vector x^k, that is,

$$\psi_0 = x^k, \qquad \psi_i = \psi_{i-1} - \alpha^k \nabla g_i(x^k) g_i(x^k), \quad i = 1, \ldots, m, \tag{4}$$

so that the iteration consisting of a cycle over the entire data set starting from x^k has the form

$$x^{k+1} = x^k - \alpha^k \sum_{i=1}^{m} \nabla g_i(x^k) g_i(x^k) = x^k - \alpha^k \nabla f(x^k). \tag{5}$$

Incremental methods are supported by stochastic [PoT73], [Lju77], [KuC78], [Pol87], [BeT89], [Whi89], [Gai93], as well as deterministic convergence analyses [Luo91], [Gri93], [LuT93], [MaS93], [Man93]. It has been experimentally observed that the incremental gradient method (2)-(3) often converges much faster than the steepest descent method (5) when far from the eventual limit. However, near convergence, the incremental gradient method typically converges slowly because it requires a diminishing stepsize $\alpha^k = O(1/k)$ for convergence. If α^k is instead taken to be a small constant, an oscillation within each data cycle arises, as shown by [Luo91]. By contrast, for convergence of the steepest descent method, it is sufficient that the stepsize α^k is a small constant (this requires that ∇g_i be Lipschitz continuous, see e.g. [Pol87]). The asymptotic convergence rate of steepest descent with a constant stepsize is typically linear and much faster than that of the incremental gradient method.

The behavior described above is most vividly illustrated in the case where the data blocks are linear and the vector x is one-dimensional, as shown in the following example:

Example 1:

Consider the least squares problem

$$\begin{aligned} &\text{minimize} \quad f(x) = \frac{1}{2} \sum_{i=1}^{m} (a_i x - b_i)^2 \\ &\text{subject to} \quad x \in \Re, \end{aligned} \tag{6}$$

where a_i and b_i are given scalars with $a_i \neq 0$ for all i. The minimum of each of the squared data blocks

$$f_i(x) = \frac{1}{2}(a_i x - b_i)^2 \tag{7}$$

is

$$x_i^* = \frac{b_i}{a_i},$$

while the minimum of the least squares cost function f is

$$x^* = \frac{\sum_{i=1}^{m} a_i b_i}{\sum_{i=1}^{m} a_i^2}.$$

It can be seen that x^* lies within the range of the data block minima

$$R = \left[\min_i x_i^*, \max_i x_i^* \right], \tag{8}$$

and that for all x *outside* the range R, the gradient

$$\nabla f_i(x) = a_i(a_i x - b_i)$$

has the same sign as $\nabla f(x)$. As a result, the incremental gradient method given by

$$\psi_i = \psi_{i-1} - \alpha^k \nabla f_i(\psi_{i-1}) \tag{9}$$

[cf. Eq. (2)], approaches x^* at each step provided the stepsize α^k is small enough. In fact it is sufficient that

$$\alpha^k \leq \min_i \frac{1}{a_i^2}. \tag{10}$$

However, for x *inside* the region R, the ith step of a cycle of the incremental gradient method, given by (9), need not make progress because it aims to approach x_i^* but not necessarily x^*. It will approach x^* (for small enough stepsize α^k) only if the current point ψ_{i-1} does not lie in the interval connecting x_i^* and x^*. This induces an oscillatory behavior within the region R, and as a result, the incremental gradient method will typically not converge to x^* unless $\alpha^k \to 0$. By contrast, it can be shown that the steepest descent method, which takes the form

$$x^{k+1} = x^k - \alpha^k \sum_{i=1}^m a_i(a_i x^k - b_i),$$

converges to x^* for any constant stepsize satisfying

$$\alpha^k \leq \frac{1}{\sum_{i=1}^m a_i^2}. \tag{11}$$

However, unless the stepsize choice is particularly favorable, for x outside the region R, a full iteration of steepest descent need not make more progress towards the solution than a single step of the incremental gradient method. In other words, *far from the solution (outside R), a single pass through the entire data set by the incremental gradient method is roughly as effective as m passes through the data set by the steepest descent method.*

The analysis of the preceding example relies on x being one-dimensional, but in many multidimensional problems the same qualitative behavior can be observed. In particular, a pass through the ith data block g_i by the incremental gradient method can make progress towards the solution in the region where the data block gradient $\nabla g_i(\psi_{i-1})g_i(\psi_{i-1})$ makes an angle less than 90 degrees with the cost function gradient $\nabla f(\psi_{i-1})$. If the data blocks g_i are not "too dissimilar", this is likely to happen in a region that is not too close to the optimal solution set. For example, consider the case where the data blocks are linear,

$$g_i(x) = z_i - C_i x,$$

where the vectors z_i and the matrices C_i are given. Then, it can be shown that sufficiently far from the optimal solution, the direction $\nabla g_i(x)g_i(x)$ used at the ith step of a data cycle

of incremental gradient will be a descent direction for the entire cost function f, if the matrix $C_i'C_i\sum_{j=1}^m C_j'C_j$ is positive definite in the sense that

$$x'C_i'C_i\left(\sum_{j=1}^m C_j'C_j\right)x > 0, \qquad \forall\, x \neq 0. \tag{12}$$

This will be true if the matrices C_i are sufficiently close to each other in terms of a matrix norm. One may also similarly argue on a heuristic basis that the incremental gradient method will be substantially more effective than the steepest descent method far from the solution if the above relation holds for a substantial majority of the indices i.

In this paper we introduce a class of gradient-like methods parameterized by a single nonnegative constant μ. For the two extreme values $\mu = 0$ and $\mu = \infty$, we obtain as special cases the incremental gradient and steepest descent methods, respectively. Positive values of μ yield hybrid methods with varying degrees of incrementalism in processing the data blocks. We also propose a time-varying hybrid method, where μ is gradually increased from $\mu = 0$ towards $\mu = \infty$. This method aims to combine the typically faster initial convergence rate of incremental gradient with the faster ultimate convergence rate of steepest descent. It starts out as the incremental gradient method (2)-(3), but gradually (based on algorithmic progress) it becomes less and less incremental, and asymptotically it approaches the steepest descent method (5). In contrast to the incremental gradient method, it uses a constant stepsize without resulting in an asymptotic oscillation. We prove convergence and a linear rate of convergence for this method in the case where the data blocks are linear. Similar results can be shown for the case of nonlinear data blocks and a parallel asynchronous computing environment. In addition to a linear convergence rate, the use of a constant stepsize offers another important practical advantage: it allows a more effective use of diagonal scaling based for example on diagonal approximations of the Hessian matrix. The convergence results suggest some practical ways of implementing the method. Our test results show a much better performance for our method than both the incremental gradient and the steepest descent method, particularly when diagonal scaling is used.

2. THE NEW INCREMENTAL GRADIENT METHOD

We embed the incremental gradient method (2)-(3) and the steepest descent method (5) within a one-parameter family of methods for the least squares problem. For a fixed $\mu \geq 0$, define

$$\xi_i(\mu) = \frac{1}{1+\mu+\cdots+\mu^{m-i}}, \qquad i = 1,\ldots,m. \tag{13}$$

Consider the method which given x^k, generates x^{k+1} according to

$$x^{k+1} = \psi_m, \tag{14}$$

where ψ_m is generated at the last step of the recursive algorithm

$$\psi_i = x^k - \alpha^k h_i, \qquad i = 1, \ldots, m, \tag{15}$$

$$h_i = \mu h_{i-1} + \sum_{j=1}^{i} \xi_j(\mu) \nabla g_j(\psi_{j-1}) g_j(\psi_{j-1}), \qquad i = 1, \ldots, m, \tag{16}$$

from the initial conditions

$$\psi_0 = x^k, \qquad h_0 = 0. \tag{17}$$

It is easily verified by induction that an alternative formula for the vectors h_i of Eq. (16) is

$$h_i = \sum_{j=1}^{i} w_{ij}(\mu) \nabla g_j(\psi_{j-1}) g_j(\psi_{j-1}), \tag{18}$$

where

$$w_{ij}(\mu) = \frac{1 + \mu + \cdots + \mu^{i-j}}{1 + \mu + \cdots + \mu^{m-j}}, \qquad i = 1, \ldots, m, \ 1 \leq j \leq i. \tag{19}$$

Since $w_{mj}(\mu) = 1$ for all j, it follows using Eqs. (15), and (18) that the vector ψ_m obtained at the end of a pass through the data blocks is

$$\psi_m = x^{k+1} = x^k - \alpha^k h_m = x^k - \alpha^k \sum_{j=1}^{m} \nabla g_j(\psi_{j-1}) g_j(\psi_{j-1}). \tag{20}$$

Note that in the special case where $\mu = 0$, we have $w_{ij}(\mu) = 1$ for all i and j, and by comparing Eqs. (15), (18), (2), and (3), we see that the method coincides with the incremental gradient method (2)-(3). In the case where $\mu \to \infty$, we have from Eqs. (15), (18), and (19), $w_{ij}(\mu) \to 0$, $h_i \to 0$, and $\psi_i \to x^k$ for $i = 0, 1, \ldots, m-1$, so by comparing Eqs. (20) and (5), we see that the method approaches the steepest descent method (5). Generally, it can be seen that as μ increases, the method becomes "less incremental".

We first prove a convergence result for the method (13)-(17) for the case where μ is fixed and the data blocks are linear. In particular, we show that if the stepsize α^k is a sufficiently small constant, the algorithm asymptotically oscillates around the optimal solution. However, the "size" of the oscillation diminishes as either $\alpha \to 0$ and μ is constant, or as α is constant and $\mu \to \infty$. If the stepsize is diminishing of the form $\alpha^k = O(1/k)$, the method converges to the minimum for all values of μ.

Proposition 1: Consider the case of linear data blocks,

$$g_i(x) = z_i - C_i x, \qquad i = 1, \ldots, m, \tag{21}$$

and the method [cf. Eq. (13)-(17)]

$$x^{k+1} = \psi_m, \tag{22}$$

where

$$\psi_0 = x^k, \qquad \psi_i = x^k - \alpha^k h_i, \quad i = 1, \ldots, m, \tag{23}$$

$$h_0 = 0, \qquad h_i = \mu h_{i-1} + \sum_{j=1}^{i} \xi_j(\mu) C_j'(C_j\psi_{j-1} - z_j), \quad i = 1, \ldots, m. \tag{24}$$

Assume that $\sum_{i=1}^m C_i'C_i$ is a positive definite matrix and let x^* be the optimal solution of the corresponding least squares problem. Then:

(a) For each $\mu \geq 0$, there exists $\overline{\alpha}(\mu) > 0$ such that if α^k is equal to some constant $\alpha \in (0, \overline{\alpha}(\mu)]$ for all k, $\{x^k\}$ converges to some vector $x(\alpha, \mu)$, and we have $\lim_{\alpha \to 0} x(\alpha, \mu) = x^*$. Furthermore, there exists $\overline{\alpha} > 0$ such that $\overline{\alpha} \leq \alpha(\mu)$ for all $\mu \geq 0$, and for all $\alpha \in (0, \overline{\alpha}]$, we have $\lim_{\mu \to \infty} x(\alpha, \mu) = x^*$.

(b) For each $\mu \geq 0$, if $\alpha^k > 0$ for all k, and

$$\sum_{k=0}^{\infty} \alpha^k = \infty, \qquad \sum_{k=0}^{\infty} (\alpha^k)^2 < \infty, \tag{25}$$

then $\{x^k\}$ converges to x^*.

Proof: (a) We first note that from Eq. (20), we have

$$x^{k+1} = x^k - \alpha \sum_{j=1}^{m} C_j'(C_j\psi_{j-1} - z_j),$$

so by using the definition $\psi_{j-1} = x^k - \alpha h_{j-1}$, we obtain

$$x^{k+1} = x^k - \alpha \sum_{j=1}^{m} C_j'(C_j x^k - z_j) + \alpha^2 \sum_{j=1}^{m} C_j'C_j h_{j-1}. \tag{26}$$

We next observe that from Eq. (18) and the definition $\psi_{j-1} = x^k - \alpha h_{j-1}$, we have for all i

$$\begin{aligned} h_i &= \sum_{j=1}^{i} w_{ij}(\mu) C_j'(C_j\psi_{j-1} - z_j) \\ &= \sum_{j=1}^{i} w_{ij}(\mu) C_j'C_j x^k - \alpha \sum_{j=1}^{i} w_{ij}(\mu) C_j'C_j h_{j-1} - \sum_{j=1}^{i} w_{ij}(\mu) C_j' z_j. \end{aligned} \tag{27}$$

From this relation, it can be seen inductively that for all i, h_i can be written as

$$h_i = \sum_{j=1}^{i} w_{ij}(\mu) C_j'C_j x^k - \sum_{j=1}^{i} w_{ij}(\mu) C_j' z_j + \alpha R_i(\alpha, \mu) x^k + \alpha r_i(\alpha, \mu), \tag{28}$$

where $R_i(\alpha,\mu)$ and $r_i(\alpha,\mu)$ are some matrices and vectors, respectively, depending on α and μ. Furthermore, using Eq. (27) and the fact that $w_{ij}(\mu) \in (0,1]$ for all i, j, and $\mu \geq 0$, we have that for any bounded interval T of stepsizes α, there exist positive uniform bounds $\overline{R}$ and $\overline{r}$ for $\|R_i(\alpha,\mu)\|$ and $\|r_i(\alpha,\mu)\|$, that is,

$$\|R_i(\alpha,\mu)\| \leq \overline{R}, \qquad \|r_i(\alpha,\mu)\| \leq \overline{r}, \qquad \forall\ i,\ \mu \geq 0,\ \alpha \in T. \tag{29}$$

From Eqs. (26), (28), and (29), we obtain

$$x^{k+1} = A(\alpha,\mu)x^k + b(\alpha,\mu), \tag{30}$$

where

$$A(\alpha,\mu) = I - \alpha\sum_{j=1}^{m} C_j'C_j + \alpha^2 S(\alpha,\mu), \tag{31}$$

$$b(\alpha,\mu) = \alpha\sum_{j=1}^{m} C_j'z_j + \alpha^2 s(\alpha,\mu), \tag{32}$$

I is the identity matrix, and the matrix $S(\alpha,\mu)$ and the vector $s(\alpha,\mu)$ are uniformly bounded over $\mu \geq 0$ and any bounded interval T of stepsizes α; that is, for some scalars $\overline{S}$ and $\overline{s}$,

$$\|S(\alpha,\mu)\| \leq \overline{S}, \qquad \|S(\alpha,\mu)\| \leq \overline{s}, \qquad \forall\ \mu \geq 0,\ \alpha \in T. \tag{33}$$

Let us choose the interval T to contain small enough stepsizes so that for all $\mu \geq 0$ and $\alpha \in I$, the eigenvalues of $A(\alpha,\mu)$ are all strictly within the unit circle; this is possible since $\sum_{j=1}^{m} C_j'C_j$ is assumed positive definite and Eqs. (31) and (33) hold. Define

$$x(\alpha,\mu) = \big(I - A(\alpha,\mu)\big)^{-1} b(\alpha,\mu). \tag{34}$$

Then $b(\alpha,\mu) = \big(I - A(\alpha,\mu)\big)x(\alpha,\mu)$, and by substituting this expression in Eq. (30), it can be seen that

$$x^{k+1} - x(\alpha,\mu) = A(\alpha,\mu)\big(x^k - x(\alpha,\mu)\big),$$

from which

$$x^{k+1} - x(\alpha,\mu) = A(\alpha,\mu)^k\big(x^0 - x(\alpha,\mu)\big), \qquad \forall\ k.$$

Since all the eigenvalues of $A(\alpha,\mu)$ are strictly within the unit circle, we have $A(\alpha,\mu)^k \to 0$, so $x^k \to x(\alpha,\mu)$.

To prove that $\lim_{\alpha\to 0} x(\alpha,\mu) = x^*$, we first calculate x^*. We set the gradient of $\|g(x)\|^2$ to zero, to obtain

$$\sum_{j=1}^{m} C_j'(C_jx^* - z_j) = 0,$$

so that

$$x^* = \left(\sum_{j=1}^m C_j' C_j\right)^{-1} \sum_{i=1}^m C_j' z_j. \tag{35}$$

Then, we use Eq. (34) to write $x(\alpha,\mu) = \big(I/\alpha - A(\alpha,\mu)/\alpha\big)^{-1}\big(b(\alpha,\mu)/\alpha\big)$, and we see from Eqs. (31) and (32) that

$$\lim_{\alpha\to 0} x(\alpha,\mu) = \left(\sum_{j=1}^m C_j' C_j\right)^{-1} \sum_{i=1}^m C_j' z_j = x^*.$$

To prove that $\lim_{\mu\to\infty} x(\alpha,\mu) = x^*$, we note that since $\lim_{\mu\to\infty} w_{ij}(\mu) = 0$ for $i = 1,\ldots,m-1$, it follows from Eqs. (27) and (28) that $\lim_{\mu\to\infty} R_i(\alpha,\mu) = 0$ and $\lim_{\mu\to\infty} r_i(\alpha,\mu) = 0$ for all $i = 1,\ldots,m-1$. Using these equations in Eq. (26), it follows that

$$\lim_{\mu\to\infty} S(\alpha,\mu) = 0, \qquad \lim_{\mu\to\infty} s(\alpha,\mu) = 0.$$

From Eqs. (31), (32), and (34), we then obtain

$$\lim_{\mu\to\infty} x(\alpha,\mu) = \left(\alpha\sum_{j=1}^m C_j' C_j\right)^{-1} \left(\alpha\sum_{j=1}^m C_j' z_j\right) = x^*.$$

(b) We need the following well-known lemma, a proof of which may be found in [Luo91].

Lemma 1: Suppose that $\{e^k\}$, $\{\gamma^k\}$, and $\{d^k\}$ are nonnegative sequences such that for all $k = 0, 1\ldots,$

$$e^{k+1} \le (1-\gamma^k)e^k + \delta^k, \qquad \gamma^k \le 1,$$

and

$$\sum_{k=0}^\infty \gamma^k = \infty, \qquad \sum_{k=0}^\infty \delta^k < \infty.$$

Then $e^k \to 0$.

From Eqs. (20), (30)-(32), we have

$$x^{k+1} = x^k - \alpha^k \sum_{j=1}^m C_j'(C_j x^k - z_j) + (\alpha^k)^2 S(\alpha^k,\mu)(x^k - x^*) + (\alpha^k)^2 e^k, \tag{36}$$

where

$$e^k = S(\alpha^k,\mu)x^* + s(\alpha^k,\mu). \tag{37}$$

Using also the expression (35) for x^*, we can write Eq. (36) as

$$x^{k+1} - x^* = \left(I - \alpha^k \sum_{j=1}^m C_j' C_j + (\alpha^k)^2 S(\alpha^k,\mu)\right)(x^k - x^*) + (\alpha^k)^2 e^k. \tag{38}$$

Let k be large enough so that a positive number γ is a lower bound to the minimum eigenvalue of $\sum_{j=1}^m C_j'C_j - \alpha^k S(\alpha^k,\mu)$, while $\alpha^k\gamma < 1$. Let also δ be an upper bound to $\|e^k\|$. Then from Eq. (38) we obtain

$$\|x^{k+1} - x^*\| \leq (1-\alpha^k\gamma)\|x^k - x^*\| + (\alpha^k)^2\delta.$$

Lemma 1, together with this relation, and the assumptions $\sum_{k=0}^\infty \alpha^k = \infty$ and $\sum_{k=0}^\infty (\alpha^k)^2 < \infty$, imply that $x^k \to x^*$. **Q.E.D.**

The following proposition shows that if μ is increased towards ∞ at a sufficiently fast rate, the sequence $\{x^k\}$ generated by the method with a constant stepsize converges at a linear rate.

Proposition 2: Suppose that in the kth iteration of the method (13)-(17), a k-dependent value of μ, say $\mu(k)$, and a constant stepsize $\alpha^k = \alpha$ are used. Under the assumptions of Prop. 1, if for some $q > 1$ and all k greater than some index $\overline{k}$, we have $\mu(k) \geq q^k$, then there exists $\overline{\alpha} > 0$ such that for all $\alpha \in (0,\overline{\alpha}]$ and k, we have $\|x^k - x^*\| \leq p(\alpha)\beta(\alpha)^k$, where $p(\alpha) > 0$ and $\beta(\alpha) \in (0,1)$ are some scalars depending on α.

Proof: We first note that if for some $q > 1$, we have $\mu(k) \geq q^k$ for k after some index $\overline{k}$, then for all $i < m$ and $j \leq i$, we have

$$w_{ij}\big(\mu(k)\big) = O(\gamma^k), \tag{39}$$

where γ is some scalar with $\gamma \in (0,1)$.

We next observe that similar to the derivation of Eq. (38), we have

$$x^{k+1} - x^* = \left(I - \alpha\sum_{j=1}^m C_j'C_j + \alpha^2 S\big(\alpha,\mu(k)\big)\right)(x^k - x^*) + \alpha^2 e^k. \tag{40}$$

where

$$e^k = S\big(\alpha,\mu(k)\big)x^* + s\big(\alpha,\mu(k)\big). \tag{41}$$

From Eq. (28), we see that h_i can be written as a finite number of terms of bounded norm, which are multiplied by some term $w_{ij}(\mu(k))$. Thus, in view of Eq. (39), for $i < m$ we have $\|h_i\| = O(\gamma^k)$. It follows that

$$\|S\big(\alpha,\mu(k)\big)\| = O(\gamma^k), \qquad \|s\big(\alpha,\mu(k)\big)\| = O(\gamma^k). \tag{42}$$

From Eq. (41) we then obtain

$$\|e^k\| = O(\gamma^k). \tag{43}$$

From Eqs. (40), (42), and (43), we obtain

$$\|x^{k+1} - x^*\| \leq \big(|1-\alpha\delta| + O(\gamma^k)\big)\|x^k - x^*\| + \alpha^2 O(\gamma^k),$$

where δ is the minimum eigenvalue of $\sum_{j=1}^m C_j' C_j$. This relation implies the desired rate of convergence result. **Q.E.D.**

There are a number of fairly straightforward extensions of the methods and the results just presented.

(1) When the data blocks are nonlinear, stationarity of the limit points of sequences $\{x^k\}$ generated by the method (13)-(17) can be shown under certain assumptions (including Lipschitz continuity of the data block gradients) for the case of a fixed μ and the stepsize $\alpha^k = \gamma/(k+1)$, where γ is a positive scalar. Contrary to the case of linear data blocks, γ may have to be chosen sufficiently small to guarantee boundedness of $\{x^k\}$. The convergence proof is similar to the one of the preceding proposition, but it is technically more involved. In the case where the stepsize is constant, $\mu \to \infty$, and the data blocks are nonlinear, it is also possible to show a result analogous to Prop. 2, but again the proof is technically complex and will not be given.

(2) Convergence results for parallel asynchronous versions of our method can be given, in the spirit of those in [TBA86], [BeT89] (Ch. 7), and [MaS93]. These results follow well-established methods of analysis that rely on the stepsize being sufficiently small.

(3) Variations of our method involving a quadratic momentum term are possible. The use of such terms dates to the heavy ball method of Polyak (see [Pol87]) in connection with the steepest descent method, and has become popular in the context of the incremental gradient method, particularly for neural network training problems (see [MaS93] for an analysis).

(4) Diagonal scaling of the gradients of the squared norm terms $\|g_i(x)\|^2$ is possible and should be helpful in many problems. Such scaling can be implemented by replacing Eq. (15) that calculates ψ_i with the equation

$$\psi_i = x^k - \alpha^k D h_i, \qquad i = 1, \ldots, m, \tag{44}$$

where D is a diagonal positive definite matrix. A common approach is to use a matrix D that is a diagonal approximation to the inverse Hessian of the cost function. For the linear least squares problem of Prop. 1, this approach uses as diagonal elements of D the inverses of the corresponding diagonal elements of the matrix $\sum_{i=1}^m C_i C_i'$. An important advantage of this type of diagonal scaling is that it simplifies the choice of a constant stepsize; a value of stepsize equal to 1 or a little smaller typically works well. Diagonal scaling is often beneficial for steepest descent-like methods that use a constant stepsize, but is not as helpful for the incremental gradient method, because the latter uses a variable

(diminishing) stepsize. For this reason diagonal scaling should be typically more effective for the constant stepsize methods proposed here than for the incremental gradient method. This was confirmed in our computational experiments. For this reason we believe that for problems where diagonal scaling is important for good performance, our constant stepsize methods have a decisive advantage over the LMS and the incremental gradient methods.

We finally note that incremental methods, including the methods proposed here, apply to cost functions that are sums of general nonlinear functions, and not just to cost functions that are the sum of squared norms.

3. IMPLEMENTATION AND EXPERIMENTATION

Let us consider algorithms where μ is iteration-dependent and is increased with k towards ∞. While Prop. 2 suggests that a linear convergence rate can be obtained by keeping α constant, we have found in our experimentation that it may be important to change α simultaneously with μ when μ is still relatively small. In particular, as the problem of Example 1 suggests, when μ is near zero and the method is similar to the incremental gradient method, the stepsize should be larger, while when μ is large, the stepsize should be of comparable magnitude to the corresponding stepsize of steepest descent.

The formula for $\xi_i(\mu)$ suggests that for $\mu \leq 1$, the incremental character of the method is strong, so we have experimented with a μ-dependent stepsize formula of the form

$$\alpha(\mu) = \begin{cases} \gamma & \text{if } \mu > 1 \\ \big(1 + \phi(\mu)\big)\gamma & \text{if } \mu \in [0,1]. \end{cases} \tag{45}$$

Here γ is the stepsize that works well with the steepest descent method, and should be determined to some extent by trial and error (if diagonal scaling is used, then a choice of γ close to 1 often works well). The function $\phi(\mu)$ is a monotonically decreasing function with

$$\phi(0) = \zeta, \qquad \phi(1) = 0, \tag{46}$$

where ζ is a scalar in the range $[0, m-1]$. Examples are

$$\phi(\mu) = \zeta(1-\mu), \qquad \phi(\mu) = \zeta(1-\mu^2), \qquad \phi(\mu) = \zeta(1-\sqrt{\mu}). \tag{47}$$

In some of the variations of the method that we experimented with, the scalar ζ was decreased by a certain factor each time μ was increased. Generally, with μ-dependent stepsize selection of

the form (45) and diagonal scaling, we have found the constant stepsize methods proposed here far more effective than the incremental gradient method that uses the same diagonal scaling and a diminishing stepsize.

Regarding the rule for increasing μ, we have experimented with schemes that start with $\mu = 0$ and update μ according to a formula of the form

$$\mu := \beta\mu + \delta,$$

where β and δ are fixed positive scalars with $\beta > 1$. The update of μ takes place at the start of a data cycle following the computation of x^{k+1} if either

$$\|x^{k+1} - x^k\| \leq \epsilon, \tag{48}$$

where ϵ is a fixed tolerance, or if $\hat{n}$ data cycles have been performed since the last update of μ, where $\hat{n}$ is chosen by trial and error. This criterion tries to update μ when the method appears to be making little further progress at the current level of μ, but also updates μ after a maximum specified number $\hat{n}$ of data cycles have been performed with the current μ.

We noted one difficulty with the method. When the number of data blocks m is large, the calculation of $\xi_i(\mu)$ using Eq. (13) involves high powers of μ. This tends to introduce substantial numerical error, when μ is substantially larger than 1. To get around this difficulty, we modified the method, by lumping together an increasing number of data blocks (the minimum number of terms in a data block was incremented by 1) each time μ was increased to a value above 1. This device effectively reduces the number of data blocks m and keeps the power μ^m bounded. In our computational experiments, it has eliminated the difficulty with numerical errors without substantially affecting the performance of the method.

REFERENCES

[BeT89] Bertsekas, D. P., and Tsitsiklis, J. N.,"Parallel and Distributed Computation: Numerical Methods," Prentice-Hall, Englewood Cliffs, N. J., 1989.

[Gai93] Gaivoronski, A. A., "Convergence Analysis of Parallel Backpropagation Algorithm for Neural Networks," Symposium on Parallel Optimization 3, Madison, July 7-9, 1993.

[Gri93] Grippo, L., "A Class of Unconstrained Minimization Methods for Neural Network Training," Symposium on Parallel Optimization 3, Madison, July 7-9, 1993.

[KuC78] Kushner, H. J., and Clark, D. S., "Stochastic Approximation Methods for Constrained and Unconstrained Systems," Springer-Verlag, NY, 1978.

[LuT93] Luo, Z. Q., and Tseng, P., "Analysis of an Approximate Gradient Projection Method with Applications to the Back Propagation Algorithm," Dept. Elec. and Computer Eng., McMaster Univ., Hamilton, Ontario and Dept. of Math., Univ. Washington, Seattle, August 1993.

[Luo91] Luo, Z. Q., "On the Convergence of the LMS Algorithm with Adaptive Learning Rate for Linear Feedforward Networks," Neural Computation, Vol. 3, 1991, pp. 226-245.

[MaS93] Mangasarian, O. L., and Solodov, M. V., "Serial and Parallel Backpropagation Convergence Via Nonmonotone Perturbed Minimization," Computer Science Dept., Computer Sciences Technical Report No. 1149, Univ. of Wisconsin-Madison, April 1993.

[Man93] Mangasarian, O. L, "Mathematical Programming in Neural Networks," ORSA J. on Computing, Vol. 5, 1993, pp. 349-360.

[Pol87] Poljak, B. T, "Introduction to Optimization," Optimization Software Inc., N.Y., 1987.

[TBA86] Tsitsiklis, J. N., Bertsekas, D. P., and Athans, M., "Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms," IEEE Trans. on Aut. Control, Vol. AC-31, 1986, pp. 803-812.

[Whi89] White, H., "Some Asymptotic Results for Learning in Single Hidden-Layer Feedforward Network Models," J. Am. Statistical Association, Vol. 84, 1989.

[WiH60] Widrow, B., and Hoff, M. E., "Adaptive Switching Circuits," Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, part 4, 1960, pp. 96-104.

[WiS85] Widrow, B., and Stearns, S. D., "Adaptive Signal Processing," Prentice-Hall, Englewood Cliffs, N. J., 1985.