

Multiplicative Updates for L_1 -Regularized Linear and Logistic Regression

Fei Sha¹, Y. Albert Park², and Lawrence K. Saul²

¹ Computer Science Division, University of California
Berkeley, CA 94720-1776

² Department of Computer Science and Engineering
UC San Diego, La Jolla, CA 92093-0404
feisha@cs.berkeley.edu, {yapark, saul}@cs.ucsd.edu

Abstract. Multiplicative update rules have proven useful in many areas of machine learning. Simple to implement, guaranteed to converge, they account in part for the widespread popularity of algorithms such as nonnegative matrix factorization and Expectation-Maximization. In this paper, we show how to derive multiplicative updates for problems in L_1 -regularized linear and logistic regression. For L_1 -regularized linear regression, the updates are derived by reformulating the required optimization as a problem in nonnegative quadratic programming (NQP). The dual of this problem, itself an instance of NQP, can also be solved using multiplicative updates; moreover, the observed duality gap can be used to bound the error of intermediate solutions. For L_1 -regularized logistic regression, we derive similar updates using an iteratively reweighted least squares approach. We present illustrative experimental results and describe efficient implementations for large-scale problems of interest (e.g., with tens of thousands of examples and over one million features).

1 Introduction

The search for sparse solutions appears as a theme in many seemingly unrelated areas of statistical learning. These areas include, for example, large margin classification by support vector machines (SVMs) [16], unsupervised learning by nonnegative matrix factorization (NMF) [8], and linear and logistic regression with L_1 -norm regularization [3, 6, 12, 15]. Between the first two of these areas, there recently emerged an unexpected connection. In particular, it was shown [13] that the main optimization in SVMs—an instance of nonnegative quadratic programming (NQP)—could be solved by a generalization of certain multiplicative updates proposed for NMF [8].

In this paper, we establish another connection in this framework. Specifically, we show that the same multiplicative updates developed for SVMs can also be used for L_1 -regularized linear and logistic regression. The advantages of multiplicative updates for learning sparse representations [5, 8] also transfer directly to this new setting.

The multiplicative updates that we study have two particularly appealing features. First, they are very simple to implement, with no tunable parameters or

ad-hoc heuristics needed to ensure convergence. Second, they provide a guarantee of monotonic convergence, decreasing their loss functions at each iteration.

The transparency and reliability of these updates make them attractive for many applications in machine learning. In many real-world applications, it is necessary to modify or monitor the core optimizations; sometimes, they must be re-implemented on different platforms or distributed across multiple processors. These needs are not well-served by complicated black-box solvers.

The multiplicative updates we study in this paper have proven useful in many settings. In their simplest form, originally derived for NMF [8], the updates have been widely adopted for unsupervised learning and feature extraction. In their more general form, originally derived for SVMs [14], they have also been applied to problems in acoustic echo cancellation [10] and astrophysical data analysis [2]. We believe that these multiplicative updates will prove similarly attractive to many practitioners of L_1 -regularized regression.

The paper is organized as follows. In section 2, we review the multiplicative updates proposed by [14] for nonnegative quadratic programming. We also show how to bound the error of intermediate solutions using ideas from convex duality. These types of guarantees have not been discussed in earlier work [13] on multiplicative updates. In sections 3 and 4, we describe how these updates are used for linear and logistic regression with L_1 -norm regularization. These applications of multiplicative updates to L_1 -norm regularized prediction also represent a novel contribution beyond previous work [13]. Finally, in section 5, we highlight several recent approaches most closely related to our own and conclude by discussing future directions for research.

2 Background in NQP

The problem of nonnegative quadratic programming (NQP) takes the form:

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{v}) = \frac{1}{2}\mathbf{v}^\top \mathbf{A}\mathbf{v} + \mathbf{b}^\top \mathbf{v} \\ \text{subject to} \quad & \mathbf{v} \geq \mathbf{0} . \end{aligned} \tag{1}$$

The notation $\mathbf{v} \geq \mathbf{0}$ is used to indicate that all the elements of \mathbf{v} are required to be nonnegative. For simplicity, we assume that the matrix \mathbf{A} in eq. (1) is strictly positive definite, and hence there exists a unique global minimum $\mathbf{v}^* \geq \mathbf{0}$ to this problem. Due to the nonnegativity constraints in eq. (1), its solution cannot be found in closed form. Thus, an iterative approach is required. Section 2.1 reviews the multiplicative updates proposed by Sha et al [13, 14] for these problems in NQP. Section 2.2 shows that multiplicative updates can also be used to solve the Lagrange dual problems, yielding bounds on the error of intermediate solutions. To our knowledge, this aspect of convex duality has not been exploited in previous work on multiplicative updates.

2.1 Multiplicative updates

The multiplicative updates for NQP are written in terms of matrices \mathbf{A}^+ and \mathbf{A}^- that store the positive and negative elements of \mathbf{A} . In particular, we define:

$$A_{ij}^+ = \begin{cases} A_{ij} & \text{if } A_{ij} > 0, \\ 0 & \text{otherwise.} \end{cases} \quad A_{ij}^- = \begin{cases} |A_{ij}| & \text{if } A_{ij} < 0, \\ 0 & \text{otherwise.} \end{cases}$$

It follows from these definitions that $\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-$. The multiplicative updates for NQP involve matrix-vector products between \mathbf{A}^+ and \mathbf{A}^- and the current estimate \mathbf{v} . As shorthand, we define vectors with elements $a_i = (\mathbf{A}^+ \mathbf{v})_i$ and $c_i = (\mathbf{A}^- \mathbf{v})_i$. In terms of the vectors $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^d$, the multiplicative updates take the simple closed form:

$$v_i \leftarrow \left[\frac{-b_i + \sqrt{b_i^2 + 4a_i c_i}}{2a_i} \right] v_i . \quad (2)$$

The updates assume that the vector \mathbf{v} is initialized with strictly positive elements. Then, as shown in [13], these updates converge monotonically to the global minimum of eq. (1), decreasing the loss function at each iteration.

Implementation of the updates is straightforward. In many applications, the updates can be performed by just a few lines of MATLAB code. Each update requires two matrix-vector multiplications for $\mathbf{A}^+ \mathbf{v}$ and $\mathbf{A}^- \mathbf{v}$, essentially twice the computation required to evaluate the gradient. When the matrix \mathbf{A} is itself nonnegative, eq. (2) reduces to previously derived updates for nonnegative matrix factorization [8], now widely used for unsupervised learning and feature extraction.

2.2 Convex duality

By minimizing the Lagrangian associated with eq. (1), we obtain the Lagrange dual function [1]:

$$g(\boldsymbol{\lambda}) = \inf_{\mathbf{v} \in \mathbb{R}^d} \left[f(\mathbf{v}) - \boldsymbol{\lambda}^\top \mathbf{v} \right]. \quad (3)$$

The Lagrange dual function can be used to obtain lower bounds on the solution of eq. (1). In particular, for any nonnegative pair $\mathbf{v}, \boldsymbol{\lambda} \geq \mathbf{0}$, we have:

$$g(\boldsymbol{\lambda}) = \inf_{\mathbf{u} \in \mathbb{R}^d} \left[f(\mathbf{u}) - \boldsymbol{\lambda}^\top \mathbf{u} \right] \leq \left[f(\mathbf{v}) - \boldsymbol{\lambda}^\top \mathbf{v} \right] \leq f(\mathbf{v}). \quad (4)$$

The tightest lower bound in eq. (4) is obtained by maximizing the Lagrange dual function over $\boldsymbol{\lambda} \geq \mathbf{0}$. For the NQP in eq. (1), assuming the matrix \mathbf{A} is invertible, the Lagrange dual problem is given by:

$$\begin{aligned} & \text{Maximize} && g(\boldsymbol{\lambda}) = -\frac{1}{2} (\boldsymbol{\lambda} - \mathbf{b})^\top \mathbf{A}^{-1} (\boldsymbol{\lambda} - \mathbf{b}) \\ & \text{subject to} && \boldsymbol{\lambda} \geq \mathbf{0} . \end{aligned} \quad (5)$$

Note that the Lagrange dual problem is also an instance of NQP, whose global maximum $\boldsymbol{\lambda}^* \geq \mathbf{0}$ can be computed using multiplicative updates. Finally, we note that all problems in NQP exhibit strong duality [1], guaranteeing that $g(\boldsymbol{\lambda}^*) = f(\mathbf{v}^*)$. By solving primal and dual problems in parallel, we can therefore bound the error of intermediate solutions that are obtained from multiplicative updates. We will use the observed duality gap $[f(\mathbf{v}) - g(\boldsymbol{\lambda})]$ for this purpose in section 3, when we reformulate L_1 -regularized linear regression as an instance of NQP. For example, Fig. 1 shows the observed duality gap as a function of the number of multiplicative updates.

3 L_1 -regularized linear regression

In this section, we show how to use the multiplicative updates in eq. (2) for the problem of linear regression with L_1 -norm regularization [3, 15]. The training data for linear regression consists of labeled examples $\{(\mathbf{x}_\alpha, y_\alpha)\}_{\alpha=1}^n$, where $\mathbf{x}_\alpha \in \mathbb{R}^d$ and $y_\alpha \in \mathfrak{R}$. The L_1 -regularized loss is given by:

$$\text{LOSS}(\mathbf{w}) = \frac{1}{2n} \sum_{\alpha=1}^n (y_\alpha - \mathbf{w}^\top \mathbf{x}_\alpha)^2 + \gamma \sum_{i=1}^d |w_i|, \quad (6)$$

where $\mathbf{w} \in \mathbb{R}^d$ is the weight vector and $\gamma \geq 0$ is the regularization parameter. Let \mathbf{w}^* denote the weight vector that minimizes the L_1 -regularized loss. The second term on the right hand side of eq. (6) encourages sparse solutions to this problem; in particular, larger values of γ lead to increasing numbers of zeros among the elements of \mathbf{w}^* .

To cast L_1 -regularized linear regression as an instance of NQP, specifically in the form of eq. (1), we define: $\mathbf{A} = \frac{1}{n} \sum_{\alpha=1}^n \mathbf{x}_\alpha \mathbf{x}_\alpha^\top$ and $\mathbf{b} = -\frac{1}{n} \sum_{\alpha=1}^n y_\alpha \mathbf{x}_\alpha$. With these definitions, we can rewrite the L_1 -regularized loss in eq. (6) up to an additive constant as:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2} \mathbf{w}^\top \mathbf{A} \mathbf{w} + \mathbf{b}^\top \mathbf{w} + \gamma \sum_{i=1}^d |w_i| \quad (7)$$

Section 3.1 describes how to minimize the right hand side of eq. (7) by solving a problem in NQP. Section 3.2 derives the special structure of the Lagrange dual problem for this NQP. Finally, section 3.3 presents several illustrative experimental results.

3.1 Primal formulation as NQP

We reformulate the optimization of the L_1 -regularized loss by decomposing \mathbf{w} into its positive and negative components. In particular, we introduce nonnegative variables \mathbf{u} and \mathbf{v} such that:

$$\mathbf{w} = \mathbf{u} - \mathbf{v}, \quad \mathbf{u} \geq \mathbf{0}, \quad \mathbf{v} \geq \mathbf{0} . \quad (8)$$

As shorthand notation, we also let $\gamma \in \mathfrak{R}^d$ denote the vector whose every element is equal to the scalar regularizer γ in eq. (7). Finally, in terms of the variables \mathbf{u} and \mathbf{v} , we consider the optimization:

$$\begin{aligned} \text{Minimize } & f(\mathbf{u}, \mathbf{v}) = \frac{1}{2}(\mathbf{u}-\mathbf{v})^\top \mathbf{A}(\mathbf{u}-\mathbf{v}) + \mathbf{b}^\top(\mathbf{u}-\mathbf{v}) + \gamma^\top(\mathbf{u}+\mathbf{v}) \\ \text{subject to } & \mathbf{u} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0} . \end{aligned} \quad (9)$$

Let $\begin{bmatrix} \mathbf{u}^* \\ \mathbf{v}^* \end{bmatrix}$ denote the minimizing solution of eq. (9). It is straightforward to show that either $u_i^* = 0$ and/or $v_i^* = 0$ at this minimum, due to the effect of the regularizer $\gamma^\top(\mathbf{u} + \mathbf{v})$. It follows that $u_i^* + v_i^* = |u_i^* - v_i^*|$, and hence the minimum of eq. (9) maps directly onto the minimum of eq. (7). Thus we can use one problem to solve the other.

The change of variables in eq. (8) follows the strategy suggested by Koh et al [6], transforming the non-differentiable objective function in eq. (7) to the differentiable one in eq. (9). Here, the change of variables casts the problem of L_1 -regularized linear regression as an instance of NQP. The NQP problem in eq. (9) can be solved using the multiplicative updates from section 2. Note that in this context, the updates are applied to the $2d$ -dimensional nonnegative vector $\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}$ obtained by concatenating the elements of $\mathbf{u} \in \mathfrak{R}_+^d$ and $\mathbf{v} \in \mathfrak{R}_+^d$.

3.2 Dual formulation as BQP

By minimizing the Lagrangian associated with eq. (9), we obtain the Lagrange dual function:

$$g(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \inf_{\mathbf{u}, \mathbf{v} \in \mathfrak{R}^d} \left[f(\mathbf{u}, \mathbf{v}) - \boldsymbol{\theta}^\top \mathbf{u} - \boldsymbol{\lambda}^\top \mathbf{v} \right] \quad (10)$$

In general, the right hand side of eq. (10) is unbounded below, as can be seen by evaluating it in the special case that $\mathbf{u} = \mathbf{v}$:

$$\left[f(\mathbf{v}, \mathbf{v}) - \boldsymbol{\theta}^\top \mathbf{v} - \boldsymbol{\lambda}^\top \mathbf{v} \right] = (2\gamma - \boldsymbol{\theta} - \boldsymbol{\lambda})^\top \mathbf{v} \quad (11)$$

Setting $\mathbf{u} = \mathbf{v}$, the right hand side of eq. (10) thus reduces to a single linear term in \mathbf{v} . A finite minimum does not exist because we can scale the magnitude of \mathbf{v} in eq. (11) to be arbitrarily large. The Lagrange dual function $g(\boldsymbol{\theta}, \boldsymbol{\lambda})$ does, however, have a well-defined minimum in the special case that $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ are chosen precisely to cancel the divergence in eq. (11). In particular, suppose that:

$$\boldsymbol{\lambda} + \boldsymbol{\theta} = 2\gamma, \quad (12)$$

which causes the right hand side of eq. (11) to vanish. Enforcing this constraint, and substituting $f(\mathbf{u}, \mathbf{v})$ from eq. (9) into eq. (10), we find that the variables \mathbf{u} and \mathbf{v} appear in the Lagrangian only through their difference $\mathbf{w} = \mathbf{u} - \mathbf{v}$. In particular, with these substitutions, the minimization in eq. (10) reduces to:

$$g(\boldsymbol{\theta}, \boldsymbol{\lambda})|_{\boldsymbol{\lambda}+\boldsymbol{\theta}=2\gamma} = \inf_{\mathbf{w} \in \mathfrak{R}^d} \left[\frac{1}{2} \mathbf{w}^\top \mathbf{A} \mathbf{w} + \frac{1}{2} (2\mathbf{b} + \boldsymbol{\lambda} - \boldsymbol{\theta})^\top \mathbf{w} \right]. \quad (13)$$

The quadratic form in eq. (13) yields a simple minimization. Thus, collecting the two different regimes (bounded and unbounded) of the dual, we have:

$$g(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \begin{cases} -\frac{1}{8} (2\mathbf{b} + \boldsymbol{\lambda} - \boldsymbol{\theta})^\top \mathbf{A}^{-1} (2\mathbf{b} + \boldsymbol{\lambda} - \boldsymbol{\theta}) & \text{if } \boldsymbol{\lambda} + \boldsymbol{\theta} = 2\boldsymbol{\gamma}, \\ -\infty & \text{otherwise.} \end{cases} \quad (14)$$

As we shall see, the existence of these different regimes gives rise to a Lagrange dual problem with more structure than generic instances of NQP.

We can maximize the Lagrange dual function $g(\boldsymbol{\theta}, \boldsymbol{\lambda})$ over all nonnegative $\boldsymbol{\theta}, \boldsymbol{\lambda} \geq \mathbf{0}$ to derive a lower bound on the primal optimization in eq. (9). Clearly, for this maximization we only need to consider the domain over which the function $g(\boldsymbol{\theta}, \boldsymbol{\lambda})$ is finite and bounded below. Over this domain, we can use the equality constraint in eq. (12) to eliminate the variable $\boldsymbol{\lambda}$ and obtain the simpler maximization:

$$\begin{aligned} &\text{Maximize } h(\boldsymbol{\theta}) = -\frac{1}{2} (\boldsymbol{\theta} - \mathbf{b} - \boldsymbol{\gamma})^\top \mathbf{A}^{-1} (\boldsymbol{\theta} - \mathbf{b} - \boldsymbol{\gamma}) \\ &\text{subject to } \mathbf{0} \leq \boldsymbol{\theta} \leq 2\boldsymbol{\gamma}. \end{aligned} \quad (15)$$

This optimization is an instance of box quadratic programming (BQP), since in addition to the nonnegativity constraint on $\boldsymbol{\theta}$, there also appears the box constraint implied by eq. (12). The optimization can be solved using a variant of the multiplicative updates reviewed in section 2. The updates for BQP include a clipping operation to enforce the upper bound $\boldsymbol{\theta} \leq 2\boldsymbol{\gamma}$; for further details, see [13, 14].

Note how the special structure of the primal optimization in eq. (9) is manifested in its dual Lagrange problem, eq. (15). In general, as shown in section 2, primal optimizations in NQP have dual optimizations in NQP, with both optimizations over variables of the same dimensionality. Here, however, the NQP in eq. (9) over the joint variables $\mathbf{u}, \mathbf{v} \in \mathbb{R}_+^d$ generates as its Lagrange dual problem a smaller instance of BQP over the single variable $\boldsymbol{\theta} \in \mathbb{R}_+^d$.

3.3 Experimental results

We experimented with the multiplicative updates for NQP to investigate their performance on problems in L_1 -regularized linear regression. For these experiments, we created artificial data sets $\{(\mathbf{x}_\alpha, y_\alpha)\}_{\alpha=1}^n$ with inputs of varying dimensionality $d \sim 10^{2-3}$.

Each data set was created as follows. First, we randomly generated a “ground truth” weight vector $\mathbf{w} \in \mathbb{R}^d$ with precisely $d/3$ negative elements, $d/3$ zero elements, and $d/3$ positive elements. The magnitudes of nonzero elements in this weight vector were sampled uniformly from the unit interval. For ease of visualization, the elements of \mathbf{w} were also sorted from smallest to largest. (See the top left panel of Fig. 2.) Second, we randomly generated n inputs by sampling the elements of each input vector \mathbf{x}_α from a zero-mean Gaussian with unit variance. Finally, we generated n outputs by sampling each y_α from a Gaussian distribution with mean $\mu_\alpha = \mathbf{w}^\top \mathbf{x}_\alpha$ and standard deviation 0.2σ , where σ measured the standard deviation of the means $\{\mu_\alpha\}_{\alpha=1}^n$.

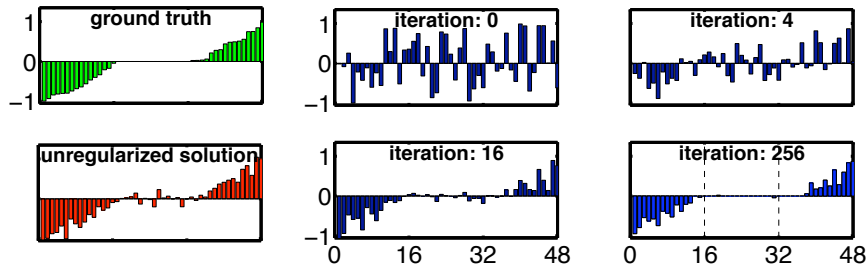


Fig. 2. Evolution of weight vector $\mathbf{w} \in \mathbb{R}^{48}$ under multiplicative updates for L_1 -regularized linear regression, starting from random initialization. Also shown are the “ground truth” weight vector (that would have been recovered in the absence of noise) and unregularized solution.

The noise in these data sets prevents a linear regression from exactly recovering the ground truth weight vector. The use of L_1 -norm regularization, however, encourages sparse solutions, so that an L_1 -regularized linear regression may be expected to yield an estimated weight vector with the same (or nearly the same) sparsity pattern. In the experiments reported below, the data sets had $n = 2d$ examples (so as to scale with the dimensionality of the inputs), and we set the regularization parameter to $\gamma = 0.1$. In this regime, L_1 -norm regularization had the desired effect of encouraging appropriately sparse solutions. Our experiments were designed to measure the convergence of the multiplicative updates in this regime.

First, we present typical results from L_1 -regularized linear regression on data with input dimensionality $d = 48$. Fig. 1 shows the observed duality gap between the primal and dual optimizations in eqs. (9) and (15) as a function of the number of multiplicative updates. Similarly, Fig. 2 shows the convergence of the weight vector $\mathbf{w} = \mathbf{u} - \mathbf{v}$ obtained from the primal optimization. For this figure, the elements of the weight vector were initialized at random. Also shown in the plot are the “ground truth” weight vector used to generate this data set, as well as the weight vector obtained from a linear regression without L_1 -norm regularization. In both figures, it can be seen that the multiplicative updates converge reliably to the global minimum.

Next we present results on problems of varying input dimensionality d . We generated random data sets (as described above) with inputs of dimensionality $d = 48, 96, 192, 384, 768,$ and 1536 . We generated twelve data sets for each input dimensionality and averaged our results across these twelve data sets. For these

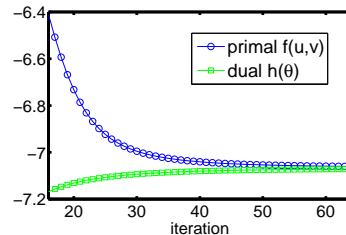


Fig. 1. Convergence of multiplicative updates for primal and dual optimizations in L_1 -regularized linear regression; see text in section 3.3 for details.

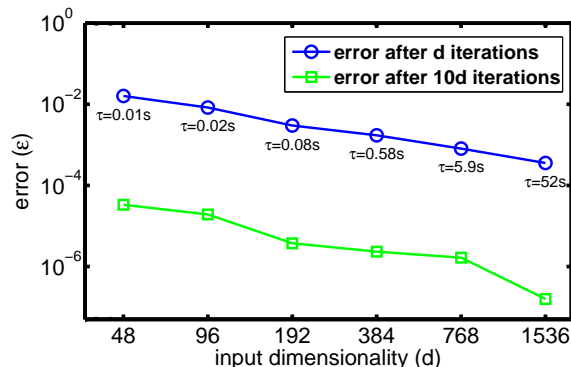


Fig. 3. Error ε_t from eq. (16) after $t = d$ and $t = 10d$ iterations of the multiplicative updates, on data sets of input dimensionality d . Each result represents an average over twelve different, randomly generated data sets. For the experiments with $t = d$ iterations, we also indicate the CPU times τ in seconds (per data set).

experiments, the weight vector was initialized by performing a linear regression without L_1 -norm regularization: namely, $\mathbf{w}_0 = \mathbf{A}^{-1}\mathbf{b}$, with \mathbf{A} and \mathbf{b} defined as in eq. (7). We measured the convergence of the multiplicative updates as follows. Let \mathbf{u}_t , \mathbf{v}_t , and $\boldsymbol{\theta}_t$ denote the vectors obtained after t updates on the primal and dual optimizations in eqs. (9) and (15). Also, let $\mathbf{w}_t = \mathbf{u}_t - \mathbf{v}_t$. We computed the error ratio:

$$\varepsilon_t = \frac{\mathcal{L}(\mathbf{w}_t) - h(\boldsymbol{\theta}_t)}{\mathcal{L}(\mathbf{w}_0) - h(\boldsymbol{\theta}_t)}. \quad (16)$$

The numerator in eq. (16) simply measures the observed duality gap, while the ratio provides an easily computed upper bound on the amount of relative improvement $\frac{\mathcal{L}(\mathbf{w}_t) - \mathcal{L}(\mathbf{w}^*)}{\mathcal{L}(\mathbf{w}_0) - \mathcal{L}(\mathbf{w}^*)}$. Note that computing eq. (16) does not require exact knowledge of $\mathcal{L}(\mathbf{w}^*)$. We report the ratio in eq. (16), as opposed to the absolute value of the duality gap, because it normalizes to some extent for the degree of regularization and the corresponding difficulty of the optimization.

The results of these experiments are summarized in Fig. 3. On data sets of varying dimensionality d , the figure shows the error ε_t after different numbers of iterations t . The results in the figure were averaged over twelve randomly generated data sets. The figure shows the average error ε_t after $t = d$ and $t = 10d$ iterations of the multiplicative updates: that is, after a number of iterations equal to and ten times greater than the input dimensionality. Again, it is seen that the multiplicative updates converge reliably and quickly to the global minimum. In terms of computation, each iteration of the updates involves four matrix-vector multiplications (two for the primal, two for the dual), but no matrix inversions or matrix-matrix multiplications. The figure also shows representative CPU times τ per data set, in MATLAB, on a Mac Pro workstation with a 2×3 GHz Dual-Core Intel Xeon processor. For t iterations of the updates, we expect $\tau = O(td^2)$, which is generally observed for medium to large values of d .

4 L_1 -regularized logistic regression

In this section, we show how to use multiplicative updates for L_1 -regularized logistic regression. The training data consists of labeled examples $\{(\mathbf{x}_\alpha, y_\alpha)\}_{\alpha=1}^n$, where $\mathbf{x}_\alpha \in \mathbb{R}^d$ and $y_\alpha \in \{0, 1\}$. Let $s_\alpha = 2y_\alpha - 1$ denote the negatively and positively labeled examples by their signs $\{-1, +1\}$. The L_1 -regularized log-loss is given by:

$$\mathcal{L}(\mathbf{w}) = -\frac{1}{n} \sum_{\alpha=1}^n \log \sigma(s_\alpha \mathbf{w}^\top \mathbf{x}_\alpha) + \gamma \sum_{i=1}^d |w_i|, \quad (17)$$

where $\mathbf{w} \in \mathbb{R}^d$ is the weight vector, $\sigma(z) = [1 + e^{-z}]^{-1}$ is the logistic function, and $\gamma \geq 0$ is the regularization parameter.

Many algorithms for logistic regression are based on local quadratic approximations [9] to the log-loss in the neighborhood of the current estimate \mathbf{w} . These quadratic approximations generate iteratively reweighted least squares (IRLS) sub-problems that can be solved using simpler methods. The simplest quadratic approximation is obtained by expanding the first term on the right hand side in eq. (17) by its Taylor series. In our work, we use a different quadratic approximation that instead provides a provably global upper bound on the log-loss. Our approximation relies on an inequality originally introduced in the context of Bayesian logistic regression [4]:

$$\log \sigma(z') \geq \log \sigma(z) + \frac{1}{2}(z' - z) - \frac{\tanh(z/2)}{4z}(z'^2 - z^2). \quad (18)$$

Eq. (18) holds for all real-valued pairs (z, z') and reduces to an equality for $z' = z$. Fig. 4 illustrates the bound around the point $z = -\frac{5}{4}$. This bound provides a more controlled approximation to the loss function than the one obtained from its Taylor series: in particular, looseness in the bound leads us only to *underestimate* the progress made in optimizing the true loss function. Applying the quadratic bound in eq. (18) to eq. (17) in the neighborhood of the current estimate \mathbf{w} , we generate a L_1 -regularized least squares sub-problem that can be solved using the multiplicative updates from section 3. In fact, it is not necessary to iterate the multiplicative updates on these least squares sub-problems to convergence. Instead, we perform just one multiplicative update, then recompute the local quadratic approximation before performing the next one.

We experimented with the 20 NEWSGROUPS data set, currently available at the web site <http://people.csail.mit.edu/jrennie/20Newsgroups>. We attempted

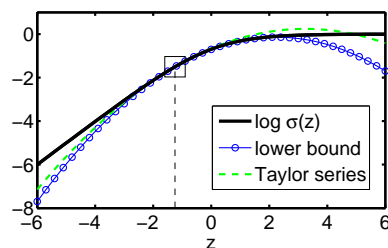


Fig. 4. Comparison of $\log \sigma(z)$, its quadratic Taylor series approximation, and the lower bound in eq. (18). The three curves meet inside the square box. Note that the Taylor series approximation does not provide a lower bound.

regularizer (γ)	sparsity (%)	time (sec)	regularizer (γ)	sparsity (%)	time (sec)
0.010	89.76	1296	0.100	93.25	986
0.025	92.24	1436	0.250	93.45	672
0.050	92.92	1218	0.500	93.54	535

Table 1. Percentage of zero elements in the weight vector and CPU times for L_1 -regularized logistic regression on the NEWGROUP data set.

to replicate the set-up in [6]. Classifiers were trained to distinguish documents from newsgroups with names of the form `sci.*`, `comp.*`, and `misc.forsale` versus all the rest. We used the Bow toolkit [11] with the command “rainbow -g 3 -h -s -O 2 -i” to generate document feature vectors. This created 18,792 examples, each with 1,258,799 features. To manage the extremely high dimensionality of the feature space, we exploited the nonnegativity and sparseness of the feature vectors. In particular, by careful bookkeeping, it is possible to implement the multiplicative updates for L_1 -regularized linear regression without having to construct or store the matrix \mathbf{A} in eq. (7). Also, we only used multiplicative updates to solve the primal optimizations for L_1 -regularized IRLS sub-problems: the dual optimizations were not constructed. The weight vectors were initialized from the results of unregularized logistic regressions, performed by limited-memory quasi-Newton methods (L-BFGS). For the L_1 -regularized solutions, we terminated the multiplicative updates when the relative improvement of eq. (17) per iteration was less than 10^{-3} or when the training error dropped below 0.01%. Though not particularly stringent, these stopping criteria sufficed for the multiplicative updates to generate sparse solutions.

Table 1 shows results averaged over eight random 70/20/10 splits of the data into training, test, and development sets. The development set in each split was used to tune the value of the regularization parameter, γ . As expected, increasing values γ led to solutions of increasing sparsity. These solutions were found in approximately 10-20 minutes of CPU time, demonstrating the feasibility of our approach for large-scale problems.

In half of the train/test/development splits of the NEWGROUP data set, we observed that regularization led to improved generalization. For a typical one of these splits, Fig. 5 shows the error rates on the test and development sets as a function of the regularization parameter γ . It is possible that more consistent improvement across splits would have been observed using a finer search for the regularization parameter γ .

5 Discussion

There is a large literature on algorithms for L_1 -regularized linear and logistic regression. Here, we highlight several recent approaches related to our own, as well as indicating important differences.

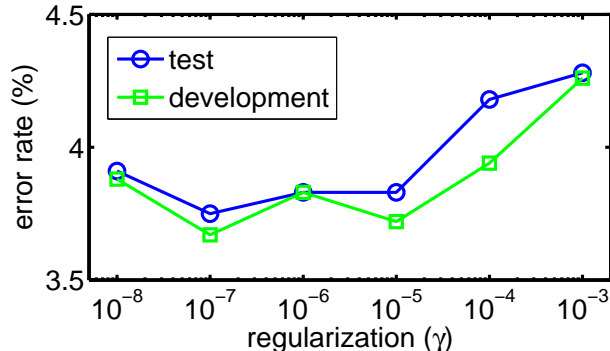


Fig. 5. Development and test error rates on one particular random split of the NEWSGROUP data. A value $\gamma = 10^{-7}$ is found that improves generalization in this split.

Closest in spirit to our approach are other bound optimization algorithms [7, 10] which derive updates from an auxiliary function. In contrast to our approach, however, these other algorithms have non-multiplicative updates that involve matrix inverses. These matrix inverses are needed to re-estimate all of the elements of the weight vector $\mathbf{w} \in \mathbb{R}^d$ in parallel (as opposed to simply performing coordinate descent). For large d , however, these matrix inverses may be prohibitively expensive.

Lee et al [9] pursue an iterative strategy for logistic regression that, like ours, is also based on quadratic approximations to the differentiable part of the log-loss. They use the LARS algorithm [3] to solve L_1 -regularized least squares problems generated by second-order Taylor expansions. Our approach differs in using a variational bound for the quadratic approximation (see Fig. 4) and calling the multiplicative updates to solve the L_1 -regularized least squares problems.

Koh et al [6] describe a fast, state-of-the-art interior point method [17] for L_1 -regularized logistic regression. Our approach was directly inspired by two aspects of their work: first, the way they recast the L_1 -norm regularizer as part of a differentiable objective function, and second, their impressive results on large-scale problems. (The NEWSGROUP data set analyzed in [6] is far larger than any of the data sets analyzed in [7, 9, 10].) Our approach differs from Koh et al [6] in that it provides a reasonably effective yet simpler re-estimation procedure. The multiplicative updates appear easier to implement, though not as fast to converge.

There are several important directions for future work. Though we have demonstrated the feasibility of our approach on a very large problem in logistic regression, with over one million features, further benchmarking is clearly needed. Also, we hope to study settings in which the updates are not merely used to solve isolated problems in L_1 -regularized regression, but are embedded in larger models with more expressive power.

Acknowledgments

We are grateful to J. Blitzer (U. of Pennsylvania) for helping us to process the NEWSGROUP data set. This work was supported by NSF Award 0238323.

References

1. S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
2. J. M. Diego, M. Tegmark, P. Protopapas, and H. B. Sandvik. Combined reconstruction of weak and strong lensing data with WSLAP, 2007. doi:10.1111/j.1365-2966.2007.11380.x.
3. B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.
4. T. Jaakkola and M. I. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10:25–37, 2000.
5. J. Kivinen and M. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.
6. K. Koh, S.-J. Kim, and S. P. Boyd. An interior-point method for large-scale ℓ_1 -regularized logistic regression. *Journal of Machine Learning Research*, 2006. Submitted for publication.
7. B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink. Sparse multinomial logistic regression: fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis and Intelligence*, 27(6):957–968, 2005.
8. D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562, Cambridge, MA, 2001. MIT Press.
9. S. Lee, H. Lee, P. Abbeel, and A. Y. Ng. Efficient ℓ_1 regularized logistic regression. In *Proceedings of the Twenty First National Conference on Artificial Intelligence*, Boston, MA, 2006.
10. Y. Lin and D. D. Lee. Bayesian ℓ_1 -norm sparse learning. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP-06)*, volume V, pages 605–608, Toulouse, France, 2006.
11. A. K. McCallum. Bow: a toolkit for statistical language modeling, text retrieval, classification and clustering. <http://www.cs.cmu.edu/mccallum/bow>, 1996.
12. A. Y. Ng. Feature selection, ℓ_1 vs. ℓ_2 regularization, and rotational invariance. In *Proceedings of the Twenty First International Conference on Machine Learning (ICML-04)*, pages 78–85, Banff, Canada, 2004.
13. F. Sha, Y. Lin, L. K. Saul, and D. D. Lee. Multiplicative updates for nonnegative quadratic programming. *Neural Computation*, 2007. Accepted for publication.
14. F. Sha, L. K. Saul, and D. D. Lee. Multiplicative updates for large margin classifiers. In *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory (COLT-03)*, pages 188–202, Washington D.C., 2003.
15. R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society B*, 58(1):267–288, 1996.
16. V. Vapnik. *Statistical Learning Theory*. Wiley, N.Y., 1998.
17. S. J. Wright. *Primal-Dual Interior Point Methods*. SIAM, Philadelphia, PA, 1997.