Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms

Tong Zhang TZHANG@WATSON.IBM.COM
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598

Abstract

Linear prediction methods, such as least squares for regression, logistic regression and support vector machines for classification, have been extensively used in statistics and machine learning. In this paper, we study stochastic gradient descent (SGD) algorithms on regularized forms of linear prediction methods. This class of methods, related to online algorithms such as perceptron, are both efficient and very simple to implement. We obtain numerical rate of convergence for such algorithms, and discuss its implications. Experiments on text data will be provided to demonstrate numerical and statistical consequences of our theoretical findings.

1. Introduction

Consider the problem of predicting an unobserved output value y based on an observed input vector x. We assume that the quality of a predictor $p(x)$ is measured by a loss function $\phi(p(x), y)$, and the data (X, Y) are drawn from an unknown underlying distribution D. Our goal is to find $p(x)$ so that the expected true loss of p given below is as small as possible:

$$Q(p(\cdot)) = E_{X,Y}\phi(p(X), Y), \qquad (1)$$

where we use $E_{X,Y}$ to denote the expectation with respect to the true (but considered to be unknown for learning problems) underlying distribution D. Throughout the paper, we assume that the loss function $\phi(p, y)$ is a convex function of p.

In this paper, we focus on linear predictors which take the form $p(x) = w^T x$. We are interested in analyzing the behavior of some simple stochastic gradient descent algorithms. This class of methods, also referred to as stochastic approximation algorithms (Kushner & Yin, 1997), have been extensively applied to learning problems such as neural networks. Similar to the perceptron method (Rosenblatt, 1962), stochastic gradient descent algorithms update the weight vector w in the online setting. We assume that training samples $(X_1, Y_1), \ldots$, are given one at a time. The algorithms examine the current data-point, and then update the weight vector accordingly. They are particularly suitable for large scale applications, where the number of data points and the problem dimensionality are both very large. Moreover, as we shall see later, the optimal prediction performance can be achieved with only a small number of iterations over the training data. Therefore SGD methods can be very efficient.

This paper focuses on the numerical aspect of SGD and its statistical consequences. In statistical learning, we want to find a predictor with small true risk in (1) using a finite number of training samples drawn from the underlying distribution D. Therefore the purpose of our analysis is to understand the convergence behavior of SGD when we apply it to the random training samples. In practice, it is helpful to run the method over the training data multiple times. The reason, based on a relationship of early-stopping and regularization, will become clear later. Another issue we investigate is the effect of predictor averaging. In the stochastic approximation literature, the averaging technique comes with great theoretical promises (Polyak & Juditsky, 1992). Our consideration mainly follows the point of view from the online learning literature, where averaging is a technical tool to derive non-asymptotic convergence bounds. In this context, we shall discuss the effect of averaging based on statistical implications of our bounds and an experimental study.

2. Stochastic Gradient Descent Method

One issue of equation (1) is that the solution may not be unique, or may not even exist (when the minimum is achieved at infinity). Practitioners often use a reg-

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004. Copyright 2004 by the first author.

ularized form such as the generalization below, which always has a unique and numerically stable solution:

$$Q_\lambda(w) = E_{X,Y}\phi(w^T X, Y) + \frac{\lambda}{2}\|w\|_2^2, \qquad (2)$$

where $\|w\|_2^2 = w^T w$ and λ is a non-negative regularization parameter. If $\lambda = 0$, the problem is unregularized.

We solve (2) using stochastic gradient descent. At each step t, we observe (X_t, Y_t), and update the current weight vector $\hat{w}_{t-1}$ using the formula $\hat{w}_t = \hat{w}_{t-1} - \eta_t S_t^{-1} \frac{\partial}{\partial w} L_\phi(\hat{w}_{t-1}, X_t, Y_t)$, where $\eta_t > 0$ is an appropriately chosen learning rate parameter, S_t is a symmetric positive definite matrix, and $L_\phi(w, x, y) = \phi(w^T x, y) + \frac{\lambda}{2}\|w\|_2^2$. The matrix S_t can be regarded as a preconditioner, which accelerates the convergence rate if chosen appropriately. In practice, one can simply let S_t be the identity matrix. For simplicity, we assume that S_t is a constant matrix. The corresponding stochastic gradient descent update rule becomes

$$\hat{w}_t = \hat{w}_{t-1} - \eta_t S^{-1}(\lambda \hat{w}_{t-1} + \phi_1'(w_{t-1}^T X_t, Y_t) X_t), \quad (3)$$

where $\phi_1'(p, y) = \frac{\partial}{\partial p}\phi(p, y)$. The algorithm, which solves (2), can be summarized below.

Algorithm 2.1 *(standard SGD)*

```
Initialize $\hat{w}_0$
for $t = 1, 2, \ldots$
    Draw $(X_t, Y_t)$ randomly from $D$.
    Update $\hat{w}_{t-1}$ as:
    $\hat{w}_t = \hat{w}_{t-1} - \eta_t S^{-1}(\lambda \hat{w}_{t-1} + \phi_1'(w_{t-1}^T X_t, Y_t) X_t)$
end
```

The online stochastic gradient method can be compared to batch learning, which approximately solves (2) by replacing the true expectation with empirical samples $(X_1, Y_1), \ldots, (X_n, Y_n)$:

$$\hat{w} = \arg\min_w \left[\frac{1}{n}\sum_{i=1}^n \phi(w^T X_i, Y_i) + \frac{\lambda}{2}\|w\|_2^2 \right]. \qquad (4)$$

Both online stochastic gradient algorithm (3) and batch learning algorithm (4) approximately solve (2). The stochastic gradient method is clearly more efficient if we run the algorithm only once over the data. As we shall see later, its generalization behavior is also simple to analyze. Even if we apply the stochastic gradient method to the unregularized formulation (1) directly, it achieves an implicit regularization effect. It is worth mentioning that one can use Algorithm 2.1 to solve the batch learning system (4) by drawing samples from the empirical distribution instead, or simply run the algorithm over the empirical data multiple times.

In the following, we focus on the finite-sample convergence analysis of stochastic gradient descent, as well as its statistical consequences.

3. A One-step Difference Inequality

Although the convergence of general stochastic gradient algorithms has been well studied in the literature (for example, see (Kushner & Yin, 1997)), the existing studies are only concerned with the asymptotic convergence behavior for parametric problems. Although such analysis is very useful, it does not apply to problems where the underlying dimensionality is large compared with the number of training examples. We note that most modern learning methods (such as kernel methods or boosting) fit linear prediction models in very high dimensional spaces.

Our goal is to investigate the finite sample convergence behavior of stochastic gradient rule (3), suitable for large scale linear prediction problems. Traditionally finite sample convergence behavior has been studied in the online learning literature. In this paper, we use an idea that generalizes the proof of perceptron mistake bound to gradient descent formulation (3) as follows. Let w be an arbitrary weight vector, we consider the quantity $R_{w,t} = \|\hat{w}_t - w\|_S^2 = (\hat{w}_t - w)^T S(\hat{w}_t - w)$. The basic idea of our analysis is to show that on average, $R_{w,t}$ decreases with sufficiently small η_t when the true risk $Q_\lambda(\hat{w}_t)$ in (2) is larger than $Q_\lambda(w)$. Since $R_{w,t} \geq 0$, we can only decrease it by a finite amount. A convergence bound can then be obtained.

Elements of our proof have also been used in the standard proof of the perceptron mistake bound, as well as more recent extensions such as (Cesa-Bianchi, 1999; Kivinen & Warmuth, 2001; Kivinen et al., 2002). However, analysis presented here is more compact, and we apply the idea directly to the more general stochastic gradient descent rule (3).

Our analysis is based on the following decomposition:

$$\begin{aligned}
&\|\hat{w}_t - w\|_S^2 \\
=&\|(\hat{w}_{t-1} - w) - \eta_t S^{-1}(\lambda w_{t-1} + \phi_1'(w_{t-1}^T X_t, Y_t) X_t)\|_S^2 \\
=&\|\hat{w}_{t-1} - w\|_S^2 + \eta_t^2 \|\lambda \hat{w}_{t-1} + \phi_1'(\hat{w}_{t-1}^T X_t, Y_t) X_t\|_{S^{-1}}^2 \\
&- 2\eta_t(\lambda \hat{w}_{t-1} + \phi_1'(\hat{w}_{t-1}^T X_t, Y_t) X_t)^T(\hat{w}_{t-1} - w) \\
=&\|\hat{w}_{t-1} - w\|_S^2 + \eta_t^2 \|\lambda \hat{w}_{t-1} + \phi_1'(\hat{w}_{t-1}^T X_t, Y_t) X_t\|_{S^{-1}}^2 \\
&- 2\eta_t(L_\phi(\hat{w}_{t-1}, X_t, Y_t) - L_\phi(w, X_t, Y_t)) \\
&- \eta_t(2d_\phi(\hat{w}_{t-1}^T X_t, w^T X_t; Y_t) + \lambda\|\hat{w}_{t-1} - w\|_2^2),
\end{aligned}$$

where in the last equation, we have used the notation $d_\phi(p, q; y) = \phi(q, y) - \phi(p, y) - \phi_1'(p, y)(q - p)$. The quantity $d_\phi(p, q; y)$ is often referred to as the Bregman divergence of $\phi(p, y)$ (with respect to p). It is well known (and easy to check) that for a convex function, its Bregman divergence is always non-negative. That is, $d_\phi(\cdot) \geq 0$. Therefore the above decomposition can be simplified as the following inequality, which is the main result of this section:

$$\begin{aligned} &\|\hat{w}_t - w\|_S^2 - \|\hat{w}_{t-1} - w\|_S^2 \qquad (5)\\ \leq &\eta_t^2 (\lambda \|\hat{w}_{t-1}\|_{S^{-1}} + \|\phi_1'(\hat{w}_{t-1}^T X_t, Y_t) X_t)\|_{S^{-1}})^2 \\ &- 2\eta_t (L_\phi(\hat{w}_{t-1}, X_t, Y_t) - L_\phi(w, X_t, Y_t)). \end{aligned}$$

4. Assumption on Loss Functions

For reference purpose, we make the following assumption on the loss function:

Assumption 4.1 *Consider a loss function $\phi(p, y)$. We assume that there exists non-negative constants A and B such that $(\phi_1'(p, y))^2 \leq A\phi(p, y) + B$.*

The purpose of this assumption is to simplify our analysis. Most interesting loss functions satisfy Assumption 4.1. A notable exception is the exponential loss used in AdaBoost. The following are some common loss functions with corresponding choices of parameters A and B (which are not unique):

- Logistic: $\phi(p, y) = \ln(1 + \exp(-py))$; $A = 0$ and $B = 1$. This loss is for binary classification problems with $y \in \{\pm 1\}$.
- SVM (hinge loss): $\phi(p, y) = \max(0, 1 - py)$; $A = 0$ and $B = 1$. This loss is for binary classification problems with $y \in \{\pm 1\}$. Note that the derivative $\phi_1'(p, y)$ at $py = 1$ is not deterministic. Equation (5), thus our analysis, still holds at the point $py = 1$ with any subgradient defined as $\phi_1'(p, y) = -uy$ ($u \in [0, 1]$). However, algorithmically, it is still useful to avoid this non-deterministic problem. We thus consider a smoothed versions of SVM as alternative.
- Quadratically smoothed SVM: $\phi(p, y) = \frac{1}{2\gamma}\max(0, 1 - py)^2$ when $py \geq 1 - \gamma$ and $\phi(p, y) = 1 - \gamma/2 - py$ otherwise; $A = 0$ and $B = 1$ (another choice is $A = 2/\gamma$ and $B = 0$). This loss is for binary classification problems with $y \in \{\pm 1\}$. The parameter γ is a positive constant. When $\gamma \to 0$, it becomes the SVM loss.
- Least Squares: $\phi(p, y) = (p - y)^2$; $A = 4$ and $B = 0$. This loss is for regression problems.
- Modified Least Squares: $\phi(p, y) = \max(0, 1 - py)^2$; $A = 4$ and $B = 0$. This loss is for binary classification problems with $y \in \{\pm 1\}$, some times referred to as quadratic SVM in the literature.
- Huber loss: $\phi(p, y) = (p - y)^2$ when $|p - y| \leq 1$ and $\phi(p, y) = 2|p - y| - 1$ otherwise; $A = 0$ and $B = 4$. This loss is for regression problems.
- Modified Huber: $\phi(p, y) = \max(0, 1 - py)^2$ when $py \geq -1$ and $\phi(p, y) = -4py$ otherwise; $A = 0$ and $B = 16$. This loss is for binary classification problems with $y \in \{\pm 1\}$. It is equivalent to quadratically smoothed SVM with $\gamma = 2$.

5. Convergence Analysis of Unregularized Formulation

For the unregularized formulation, we choose $\lambda = 0$. Although (2) may not have a finite solution, $Q(\hat{w}_t)$ can still convergence to the minimum risk as $t \to \infty$.

Under Assumption 4.1, and in addition we assume that $\sup \|X_t\|_{S^{-1}} \leq M$. From (5), we obtain

$$\begin{aligned} \|\hat{w}_t - w\|_S^2 - \|\hat{w}_{t-1} - w\|_S^2 \leq &\eta_t^2 (A\phi(\hat{w}_{t-1}^T X_t, Y_t) + B) M^2 \\ - 2\eta_t (\phi(\hat{w}_{t-1}^T X_t, Y_t) - &\phi(w^T X_t, Y_t)). \qquad (6) \end{aligned}$$

5.1. Online Regret Bound

The following bound is a direct consequence of (6). Related results can be found in online learning works such as (Cesa-Bianchi, 1999; Kivinen & Warmuth, 2001).

Theorem 5.1 *Under Assumption 4.1, and in addition we assume that $\sup \|X_t\|_{S^{-1}} \leq M$. If we pick $\hat{w}_0 = 0$, $\lambda = 0$, and $\eta_t = \eta > 0$, then*

$$\begin{aligned} &(1 - \frac{\eta}{2} A M^2) \frac{1}{T} \sum_{t=1}^{T} \phi(\hat{w}_{t-1}^T X_t, Y_t) \\ \leq &\inf_w \left[\frac{1}{T} \sum_{t=1}^{T} \phi(w^T X_t, Y_t) + \frac{1}{2\eta T} \|w\|_S^2 \right] + \frac{\eta}{2} B M^2. \end{aligned}$$

Proof. Summing (6) over $t = 1, \ldots, T$ and rearranging, we obtain the desired bound. □

The above bound shows that by picking η such that η is small (but ηT is large), the average loss suffered from the online update is almost as small as that of the best regularized loss of any weight vector which is picked a priori. This applies to all loss functions that satisfy Assumption 4.1.

In the near-separable case, gradient descent with certain loss functions listed in Section 4 behave similarly as the perceptron algorithm (as far as mistake bound is concerned). We consider the quadratically smoothed SVM loss with $\gamma = 1$, and take the choice of $A = 2$

and $B = 0$. Now Theorem 5.1 implies that

$$(1 - \eta M^2)\frac{1}{T}\sum_{t=1}^{T}\phi(\hat{w}_{t-1}^T X_t, Y_t)$$
$$\leq \inf_w \left[\frac{1}{T}\sum_{t=1}^{T}\phi(w^T X_t, Y_t) + \frac{1}{2\eta T}\|w\|_S^2\right].$$

Now under the large margin assumption (required by the perceptron mistake bound), there is a weight vector w_* such that $w_*^T XY \geq 1$ with probability one. Let $\eta = 1/2M^2$ and $\hat{w}_0 = 0$, we obtain

$$2\sum_{t=1}^{T}\phi(\hat{w}_{t-1}^T X_t, Y_t) \leq 4M^2\|w_*\|_S^2.$$

Note that the left hand side is an upper bound of the mistakes the update rule makes. The right hand side is similar to the perceptron mistake bound. Therefore in the separable case, stochastic update with quadratically smoothed SVM behaves similar to the perceptron algorithm. Our bound also applies directly to the non-separable case. The corresponding result is stronger than the corresponding mistake bound for voted perceptron method in (Freund & Schapire, 1999) since in our case the convergence is with respect to the minimization of the loss function, which has important statistical consequences such as the consistency of the resulting estimator.

5.2. Convergence of Averaged Stochastic Gradient Descent

Although mistake bounds are useful by themselves, they are not the most relevant quantities to investigate if we are interested in the convergence behavior of the stochastic gradient descent rule (3) for solving the optimization problem (2). The regret bound analysis suggests us to consider the following averaged version of stochastic gradient descent algorithm. Note that we assume that we pick $\eta_t > 0$ such that $AM^2\eta_t < 2$, where $M \geq \sup \|X_t\|_{S^{-1}}$:

Algorithm 5.1 *(Averaged SGD)*

```
Initialize $\hat{w}_0$
Let $\hat{v}_0 = 0$ and $r_0 = 0$
for $t = 1, 2, \ldots$
  Draw $(X_t, Y_t)$ randomly from $D$.
  Update $\hat{w}_{t-1}$ as:
  $\hat{w}_t = \hat{w}_{t-1} - \eta_t S^{-1}(\lambda \hat{w}_{t-1} + \phi_1'(w_{t-1}^T X_t, Y_t) X_t)$
  $r_t = r_{t-1} + \eta_t - \frac{AM^2\eta_t^2}{2}$
  $\hat{v}_t = \frac{r_{t-1}}{r_t}\hat{v}_{t-1} + \frac{r_t - r_{t-1}}{r_t}\hat{w}_{t-1}$
end
```

In practical implementations of the algorithm, we choose small learning rate η_t so that $\eta_t AM^2 \ll 1$. In this case we can simply update r_t as $r_t = r_{t-1} + \eta_t$. If we pick a constant learning rate $\eta_t = \eta$, then these update rules are equivalent.

Theorem 5.2 *Under Assumption 4.1, and in addition we assume that* $\sup \|X_t\|_{S^{-1}} \leq M$. *Let* $\hat{w}_0 = 0$ *and* $\lambda = 0$, *then*

$$E\,Q(\hat{v}_T) \leq \sum_{t=1}^{T}\frac{r_t - r_{t-1}}{r_T}E\,Q(\hat{w}_{t-1})$$
$$\leq \inf_w \left[(1 + \frac{\sigma_T^2 AM^2}{2r_T})Q(w) + \frac{1}{2r_T}\|w\|_S^2\right] + \frac{\sigma_T^2}{2r_T}BM^2,$$

where $\sigma_T^2 = \sum_{t=1}^T \eta_t^2$.

Proof. The first inequality is due to the Jensen's inequality and the convexity of Q. The proof of the second inequality is similar to that of Theorem 5.1. Summing (6) over $t = 1, \ldots, T$, we obtain $\forall w$:

$$-\|w\|_S^2 \leq \sum_{t=1}^{T}\eta_t^2(A\phi(\hat{w}_{t-1}^T X_t, Y_t) + B)M^2$$
$$- 2\sum_{t=1}^{T}\eta_t(\phi(\hat{w}_{t-1}^T X_t, Y_t) - \phi(w^T X_t, Y_t)).$$

Rearranging, we get

$$\sum_{t=1}^{T}\frac{2\eta_t - \eta_t^2 A}{2r_T}\phi(\hat{w}_{t-1}^T X_t, Y_t)$$
$$\leq 2\sum_{t=1}^{T}\frac{\eta_t}{2r_T}\phi(w^T X_t, Y_t) + \frac{1}{2r_T}\|w\|_S^2 + \sum_{t=1}^{T}\frac{\eta_t^2}{2r_T}BM^2.$$

By taking expectation with respect to the random selection of data (X_t, Y_t), and noticing that $2\sum_{t=1}^T \eta_t = 2r_T + \sigma_T^2 AM^2$, we obtain the desired bound. □

This gives a bound for the averaged predictor $\hat{v}_t$. It shows that as long as we pick $\eta_t \to 0$ such that $r_t \to \infty$, then $\lim_{T\to\infty} E\,Q(\hat{v}_T) \to \inf_w Q(w)$. Moreover, if we pick $\eta_t = \eta \in (0, 2/AM^2)$ to be a constant sequence, then as $T \to \infty$,

$$E\,Q(\hat{v}_T) \leq (1 + \frac{\eta AM^2}{2})\inf_w Q(w) + \frac{\eta}{2}BM^2. \quad (7)$$

An important practical issue is how to pick a learning rate sequence $\{\eta_t\}$ which leads to fast convergence. Although in order to achieve asymptotic convergence, it is useful to pick η_t which converges to zero, some practitioners favor the simpler method of using a constant learning rate. Based on (7), such a method approximately solves the optimization problem, up to an accuracy of the order $O(\eta)$. A more relevant question is that if we want to solve the optimization problem up to a pre-determined accuracy ϵ, what is the minimum number of steps T we have to take.

Interestingly, based on the bound in Theorem 5.2, one can show that $T^{-1} = O(\epsilon^2)$. Since with constant learning rate $\eta = O(\epsilon)$, we only need $T = O(\epsilon^{-2})$ steps to achieve an accuracy of $O(\epsilon)$, we know that we cannot speed up the convergence much by using more complicated learning rate schemes (at least according to the bound in Theorem 5.2). Consider the convergence bound in Theorem 5.2. If after T steps, averaged SGD solves the optimization problem up to an accuracy of $O(\epsilon)$, then we have the conditions $\sigma_T^2/r_T = O(\epsilon)$ and $1/r_T = O(\epsilon)$ (assume either $Q(w)$ or B is not zero). For an arbitrary learning rate sequence $\{\eta_t\}$, the first inequality implies that $\sum_{t=1}^T \eta_t^2 = O(\epsilon)\sum_{t=1}^T \eta_t$. Using the Schwartz inequality, we have $\sum_{t=1}^T \eta_t^2 \geq (\sum_{t=1}^T \eta_t)^2/T$. This implies $1/T = O(\sum_{t=1}^T \eta_t^2/(\sum_{t=1}^T \eta_t)^2) = O(\epsilon/r_T) = O(\epsilon^2)$.

The above observation essentially suggests that instead of considering complicated schemes for choosing learning rate η_t, we may simply fix η_t to a predetermined small constant, and then run the SGD method until satisfactory convergence is achieved.

Another important aspect of SGD is how to choose the stopping point T. In practice, we employ SGD to solve the empirical version (4) of (2). If we go through the data only once, then based on Theorem 5.2, even without regularization, the resulting weight vector approximately solves a regularized problem with respect to the true distribution. This regularization implies that the variance of the estimator is relatively small. However, going through the data only once is likely to over-regularize the system so that the bias as indicated by the right hand side of Theorem 5.2 is unnecessarily large. If we go through the data multiple times, then the resulting weight vector approximately solves (4). Since we fit the observed data better, the bias becomes smaller. However, the variance is increased due to the additional randomness introduced from going through the data multiple times.

This analysis suggests that going through the data multiple times can be regarded as a form of bias-variance trade-off, which serves as an implicit method of regularization. Therefore choosing the stopping point T has the effect of choosing appropriate trade-off between bias and variance for optimal generalization. This phenomenon will be demonstrated by an experiment. Statistically, we have an implicit regularization scheme that is conveniently parameterized by the stopping point T, as compared with the explicit regularization scheme which parameterize the degree of regularization by λ. Numerically, this analysis suggests that it is preferable to run SGD for only a small number of iterations, instead of using it to solve (4) completely. This is an important reason why in practice SGD can be a very efficient learning method for large scale linear prediction problems.

5.3. Convergence of Standard Stochastic Gradient Descent

Although Theorem 5.2 only implies that the expected risk $EQ(\hat{v}_t)$ of the averaged predictor $\hat{v}_t$ converges, in reality, we observe that $Q(\hat{w}_t)$ also converges. Note that Theorem 5.2 essentially implies that if we randomly stop at some time $T_s \in \{1, \ldots, T\}$, then the same convergence bound on the right hand side applies to this randomly stopped estimator $\hat{w}_{T_s}$. Therefore with $T \to \infty$, we know that $E\,Q(\hat{w}_{T_s})$ converges to the optimal value if $\eta_t \to 0$.

Theoretically, an interesting problem is the behavior of $\hat{w}_T$ for an arbitrary non-randomized stopping point T. As we shall see, this problem is more easily studied under the regularization framework. However, convergence of $Q(\hat{w}_T)$ is also a consequence of Theorem 5.2. Due to the space limitation, we shall only state and prove a relatively crude version with $A = 0$ and constant learning rate $\eta_t = \eta$.

Theorem 5.3 *Under Assumption 4.1 with* $A = 0$, *and assume further that* $\sup \|X_t\|_{S^{-1}} \leq M$. *If we pick* $\hat{w}_0 = 0$, $\lambda = 0$, *and* $\eta_t = \eta$ *to be a constant, then*

$$E\,Q(\hat{w}_T) \leq \inf_w \left[Q(w) + \frac{\|w\|_S^2}{2T\eta}\right] + \eta(\frac{3}{2} + \ln(T+1))BM^2.$$

Proof Sketch. Similar to Theorem 5.2, we have (with $\hat{w}_0$ and w replaced by $\hat{w}_s$):

$$\sum_{t=s+1}^{T} \frac{1}{T-s} EQ(\hat{w}_t) \leq EQ(\hat{w}_s) + \frac{\eta(T-s+1)}{2(T-s)} BM^2.$$

Now, using induction, one obtain from this inequality:

$$\sum_{t=s+k+1}^{T} \frac{EQ(\hat{w}_t)}{T-s-k} \leq EQ(\hat{w}_s) + \eta BM^2 (1 + \sum_{j=s}^{s+k-1} \frac{1}{T-j}).$$

Now let $s + k = T - 1$, we get

$$EQ(\hat{w}_T) \leq EQ(\hat{w}_s) + \eta BM^2 \sum_{j=1}^{T-s} \frac{1}{j}.$$

Therefore, $EQ(\hat{w}_T) \leq \inf_{s \leq T} EQ(\hat{w}_s) + \eta BM^2(1 + \ln(T+1))$. Now applying Theorem 5.2. □

The factor $\ln(T+1)$ on the right hand side may be an artifact due to the simplified proof-technique used here. As pointed out earlier, with constant factor, we only need to stop at a point T where $\ln(T+1) = O(\ln \frac{1}{\eta})$. Therefore the theorem can be applied with only a loss of $O(\eta \ln \frac{1}{\eta})$ factor. The result essentially implies that even we do not choose a random stopping point, we still obtain a convergence behavior similar to that of the averaged predictor $\hat{v}_T$ in the worst case.

The issue of whether averaging helps is important in SGD algorithms. As mentioned in the introduction, the classical stochastic approximation theory such as (Polyak & Juditsky, 1992) suggests that it is possible to obtain an asymptotically optimal estimator by averaging asymptotically suboptimal estimators. Averaging is also an important technical tool in online learning since mistake bounds are obtained directly for averaged predictors. In spite of the different interpretations of averaging in different context, the fundamental role of averaging remains the same. That is, the averaged predictor is more stable than non-averaged predictors. This is why averaging may help in some cases (and not necessarily so in other cases). Due to the importance of this issue, we shall discuss it in the context of our non-asymptotic convergence analysis.

Since Theorem 5.3 is based on Theorem 5.2, the corresponding convergence bound is weaker than that of $\hat{v}_T$. However, this does not mean that the averaged predictor $\hat{v}_T$ is always superior. If we choose stopping point $T_s \in \{0, \ldots, T\}$ randomly, then based on Theorem 5.2, in order to achieve the same bound on the right hand side, we stop at a time with expected value of $T/2$. The convergence bound implies that $E\,Q(\hat{w}_{T/2})$ on average is roughly comparable to $EQ(\hat{v}_T)$, suggesting that $\hat{w}_T$ can be superior to $\hat{v}_T$. The phenomenon that the convergence behavior of $\hat{v}_T$ is similar to the convergence behavior of $\hat{w}_{T'}$ with $T' < T$ can be qualitatively observed from our experiments.

Since $\hat{w}_T$ makes a relatively large random change from $\hat{w}_{T-1}$ based on (X_T, Y_T), it has a relatively large variance. The predictor $\hat{v}_T$ reduces the variance by averaging over random samples. This implies that if we choose a large learning rate η, then because of the high variance, the averaged predictor $\hat{v}_T$ tends to be superior. If we choose a relatively small learning rate η, then due to the already low variance, the averaged predictor $\hat{v}_T$ can be inferior.

The analysis, applied to the perceptron algorithm, explains why averaged perceptron algorithm tends to help. Averaged perceptron, motivated from the voted perceptron of (Freund & Schapire, 1999), has been successfully applied to natural language learning problems (for example, see (Collins, 2002)). Note that for practical large scale applications, it is crucial to use averaging instead of voting since the latter is computationally very costly. In order to understand why averaging helps, we simply observe that the perceptron method is a special case of SGD with the SVM loss and $\eta \to \infty$. Since we are effectively using a large learning rate, the non-averaged perceptron has a lot of variance. If we use a small η in the stochastic gradient descent algorithm with the SVM loss, then averaging becomes much less helpful. This will be confirmed with experiments. Theoretically, we believe that SGD with the SVM loss and a small learning rate η can be superior to the perceptron method since we try to solve a well formed optimization problem which has important statistical consequences. Moreover, the averaged perceptron method, although useful in practice, is hard to analyze. We will see from our experiments that the theoretical advantages of SGD have observable practical consequences.

6. Parameter Convergence for the Regularized Formulation

For the regularized formulation, convergence results similar to those of the unregularized formulation can be developed. Moreover, since (2) has a unique finite solution, we can show that the weight vector $\hat{w}_t$ converges to the optimal solution. Due to the space limitation, we only discuss the convergence of the weight vector to the optimal solution in this section. For simplicity, we only state the resulting convergence bound for constant learning rate η_t.

Theorem 6.1 *Under Assumption 4.1, and in addition we assume that* $\sup \|X_t\|_{S^{-1}} \leq M$ *and* $2\lambda \leq AM^2$. *If we pick* $\hat{w}_0 = 0$, $S = I$, *and* $\eta_t = \eta > 0$ *such that* $\eta' = \eta(1 - \eta AM^2) \in [0, 1/\lambda]$, *then*

$$E\|\hat{w}_T - w_*\|_2^2 \leq (1 - \lambda\eta')^T \|w_*\|_2^2 + \frac{\eta C(1 - (1 - \lambda\eta')^{T+1})}{\lambda(1 - \eta AM^2)},$$

where w_* *is the solution of (2),* $\eta' = \eta(1 - \eta AM^2)$, *and* $C = 2(AQ_\lambda(w_*) + B)M^2$.

Proof Sketch. The convexity of $L_\phi(\cdot, x, y)$ implies that $Q_\lambda(\hat{w}_{t-1}) - Q_\lambda(w_*) \geq \frac{\lambda}{2}\|\hat{w}_{t-1} - w_*\|_2^2$. From (5), we obtain

$$\begin{aligned}
&E\left[\|\hat{w}_t - w_*\|_2^2 - \|\hat{w}_{t-1} - w_*\|_2^2\right] \\
\le& \eta^2 E\left(C'\|\hat{w}_{t-1}\|_2^2 + 2(AL_\phi(\hat{w}_{t-1}, X_t, Y_t) + B)M^2\right) \\
&- 2\eta E\left(L_\phi(\hat{w}_{t-1}, X_t, Y_t) - L_\phi(w_*, X_t, Y_t)\right) \\
\le& 2\eta^2(AQ_\lambda(w_*) + B)M^2 - \lambda\eta' E\,\|\hat{w}_{t-1} - w_*\|_2^2,
\end{aligned}$$

where $C' = \lambda(2\lambda - AM^2) \le 0$. The theorem follows by induction on t. □

The above bound shows that if we pick a sufficiently small η, and run the algorithm for a sufficiently long time t, then as $\eta \to 0$, the estimated parameter $\hat{w}_t$ converges to the true underlying parameter w_* that solves (2). This also implies that for small η, with regularization, the averaged estimator $\hat{v}_t$ will not be much better than $\hat{w}_t$ asymptotically.

7. Experiments

We investigate empirically two aspects of SGD for regularized linear systems: early stopping and averaging. Although we are mostly interested in the impact of algorithmic issues on the relative learning performance, it is still useful for us to use data such that algorithms proposed in this paper compare favorably to other learning methods. Since regularized linear classifiers achieve state of the art performance in text-categorization (Li & Yang, 2003; Zhang & Oles, 2001), we use text data in our experiments. Specifically we use the standard Mod-Apte split of the Reuters-21578 data set available from *http://www.daviddlewis.com/resources/testcollections*. In our experiments, we use word stemming without stop-word removal. Similar to the set up of (Zhang & Oles, 2001), we use the binary bag-of-word model, and select 10000 features.

In text categorization, the standard performance measures of a classification method are precision and recall instead of classification error:

$$\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \times 100$$
$$\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \times 100$$

We can also adjust the linear threshold to facilitate a trade-off between precision and recall. Following (Zhang & Oles, 2001), we report the break-even point (BEP), where precision equals recall, as an evaluation criterion for the performance of a linear classifier. The reported results are micro-averaged over all 118 non-empty categories. We run SGD (or averaged SGD) repeatedly over the data with a random ordering. By default, we choose a small constant learning rate of $\eta = 0.002$.

7.1. Early Stopping and Regularization

As pointed out in Section 5.2, early-stopping for solving the unregularized batch formulation (4) with $\lambda = 0$ has a similar effect as using explicit regularization. Figure 1 demonstrates this phenomenon qualitatively, where the least squares loss function, known to perform well for text-categorization (Li & Yang, 2003), has been used for illustration.

There are some interesting observations from this study. First, we can reach the optimal stopping point relatively quickly (about ten iterations over the data). This means that stochastic gradient algorithm is quite efficient. Secondly, averaging does not help the performance, but only smoothes the convergence curve and shifts it to the right. The smoothing is due to the variance reduction effect, and the shift is expected from the discussion in Section 5.2. Thirdly, the behavior of early stopping is rather similar to explicit regularization, as expected. However, early stopping has the advantage of being easier to implement, more efficient, and more robust.

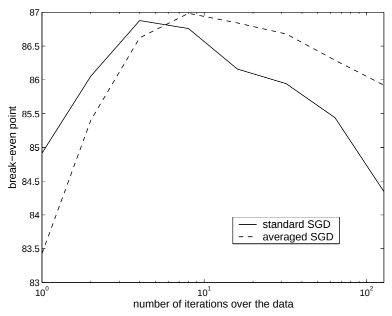

Figure 1. Performance of unregularized Least Squares versus the number of iterations over the data

7.2. Learning Rate and Averaging

Although interesting mistake bounds have been proved in (Freund & Schapire, 1999) for the voted perceptron method, the voting scheme is often too inefficient for large scale problems such as text categorization since it is infeasible to retain multiple weights. A practical remedy is to use averaging as in (Collins, 2002), which unfortunately cannot be analyzed theoretically under the perceptron framework.

In this experiment, we compare the perceptron algorithm to the stochastic gradient descent with the SVM hinge loss. As pointed out earlier, these two methods are very similar except for the following two differences. 1) with the SVM loss, we update the weight more conservatively: as long as there is a mistake within a positive margin; 2) the SGD update depends on a learning rate parameter η. The perceptron algorithm can be regarded as a special case of the SVM-loss based SGD algorithm with $\eta \to \infty$. We shall mention that the idea of including margin in a perceptron like algorithm has been considered in the online learning literature such as (Kivinen et al., 2002). However, the proposed algorithms were different from SGD for the SVM hinge loss, and this connection was not explored before. For simplicity, we shall not use explicit regularization for the SVM loss.

Figure 2 compares the performance of the perceptron method and that of SGD with the SVM loss, either with or without averaging. Perceptron based methods are clearly inferior. Again, only a small number of iterations is needed for optimal stopping. Note that the averaged perceptron and the non-averaged perceptron become similar asymptotically. This is because the training data are nearly or fully separable since the problem dimensionality (10000) is larger than the number of data points (9603). Therefore one eventually averages repeatedly over the same weight vector.

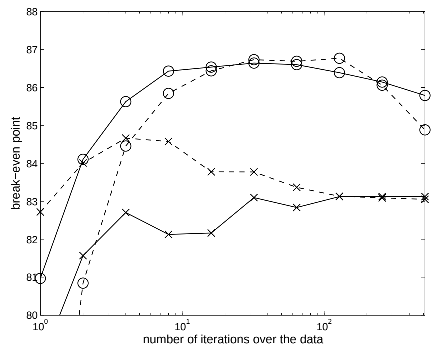

Figure 2. Performance of perceptron and SGD with SVM loss versus the number of iterations over the data. non-averaged methods: 'solid lines'; averaged methods: 'dashed lines'; perceptron: 'x', SGD : 'o'

8. Conclusion

We investigated the behavior of stochastic gradient descent methods for solving linear prediction problems. We have shown that this class of methods can be analyzed using a technique in the online learning literature that extends the traditional proof of perceptron mistake bounds. By minimizing the resulting convergence bound, we showed that the optimal convergence can be attained with a constant learning rate. Our analysis also suggests that with stochastic gradient descent, one can obtain the effect of regularization using early-stopping. Numerically, this implies that SGD is very efficient for solving large scale learning problems because we only need to run the algorithm for a small number of iterations to achieve optimal performance. This phenomenon is confirmed with experiments on text data. Moreover, we compared the proposed methods to the traditional perceptron methods, and argued that loss minimization based SGD can be superior (at least in principle). This is verified by an experiment. Our analysis also leads to useful insights into the issue of predictor averaging in online algorithms. We concluded that although useful for the perceptron method, averaging is much less important for stochastic gradient descent with a small learning rate.

References

Cesa-Bianchi, N. (1999). Analysis of two gradient-based algorithms for on-line reression. *Journal of Computer and System Sciences*, *59*, 392–411.

Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. *Proc. EMNLP'02*.

Freund, Y., & Schapire, R. (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, *37*, 277–296.

Kivinen, J., Smola, A., & Williamson, R. (2002). Large margin classification for moving targets. *Lecture Notes in Artificial Intelligence (ALT 2002)* (pp. 113–127). Springer.

Kivinen, J., & Warmuth, M. (2001). Relative loss bounds for multidimensional regression problems. *Machine Learning*, *45*, 301–329.

Kushner, H. J., & Yin, G. G. (1997). *Stochastic approximation algorithms and applications*. New York: Springer-Verlag.

Li, F., & Yang, Y. (2003). A loss function analysis for classification methods in text categorization. *ICML 03* (pp. 472–479).

Polyak, B. T., & Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, *30*, 838–855.

Rosenblatt, F. (1962). *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. New York: Spartan.

Zhang, T., & Oles, F. J. (2001). Text categorization based on regularized linear classification methods. *Information Retrieval*, *4*, 5–31.