Confidence-Weighted Sparse Online Learning

Yue Wu

November 19, 2013

1 Introduction

The main idea of Confidence-Weighted algorithms is to assume a Guassian distribution of the linear classifier $\boldsymbol{\omega} \sim N(\boldsymbol{\mu}, \Sigma)$. Each update tries to stay close to the previous distribution and ensure the probability of the current precision for $\boldsymbol{x}_i$ is larger than η.

$$Pr[y_i(\boldsymbol{\mu} \cdot \boldsymbol{x}_i) \geq 0] \geq \eta \quad or \quad y_i(\boldsymbol{\mu} \cdot \boldsymbol{x}_i) \geq \phi\sqrt{\boldsymbol{x}_i^T \Sigma \boldsymbol{x}_i} \tag{1}$$

2 Problem Formulation

In Sparse online learning, we set the weight vector equal to the average $\boldsymbol{\omega} = \boldsymbol{\mu}$. Each iteration step is a minimization problem as follows:

$$\boldsymbol{\mu}_t = \arg\min_{\boldsymbol{\mu}} f(\boldsymbol{\mu}) + \tilde{\lambda} r(\boldsymbol{\mu}) \tag{2}$$

where $f(\boldsymbol{\mu})$ is often the loss function. In this paper, we follow the setting of AROW and use the squared hinge loss:

$$f(\boldsymbol{\mu}) = \max(0, 1 - y_t(\boldsymbol{\mu_t} \cdot \boldsymbol{x}_t))^2 \tag{3}$$

To the regularization term, previous learning algorithms treat all the coordinates the same by adding an regularization term $\lambda|\omega|$. We extend the confidence to apply different regularization intensities to different coordinates. A high variance value corresponds a low confidence. Accordingly, we apply a strong shrinkage to less confident coordinates and weak shrinkage to confident ones.

2.1 Smooth Regularization

Smooth regularization is the *L1* norm. We call it smooth sparse regularization, as it shrink the weight values by some amount once a iteration.

$$r(\boldsymbol{\mu}) = |\Sigma\boldsymbol{\mu}| \tag{4}$$

2.2 Aggressive Regularization

Aggressive regularization is a strict condition on the number of non-zero coordinates, as Equ 5 shows. This setting is useful in problems like feature selection.

$$r(\boldsymbol{\mu}) = \begin{cases} 0, & |\Sigma\boldsymbol{\mu}|_0 \leq B \\ \infty, & |\Sigma\boldsymbol{\mu}|_0 > B \end{cases} \tag{5}$$

3 Solution

Common approaches such as subgradient methods to Equ. 1 will rarely lead to non-differential points of $f(\boldsymbol{\omega})$ or $r(\boldsymbol{\omega})$. While these non-differential points are the true minima in cases like *L1* regularization. Instead, we adopt a *forward-backward splitting* approach to alleviate the problems of non-differentiability.

In the first step, we adopt AROW on $f(\mu)$ to obtain one step iteration. The iteration is as Equ 6 shows.

$$\begin{aligned} \boldsymbol{\mu}_{t-\frac{1}{2}} &= \boldsymbol{\mu}_{t-1} + \alpha_t \Sigma_{t-1} y_t \boldsymbol{x}_t & \Sigma_t &= \Sigma_{t-1} - \beta_t \Sigma_{t-1} \boldsymbol{x}_t \boldsymbol{x}_t^T \Sigma_{t-1} \\ \beta_t &= \frac{1}{\boldsymbol{x}_t^T \Sigma_{t-1} \boldsymbol{x}_t + r} & \alpha_t &= \max(0, 1 - y_t \boldsymbol{x}_t^T \boldsymbol{\mu}_{t-1}) \beta_t \end{aligned} \tag{6}$$

We re-write the above iteration to be second order sub-gradient update as Equ. 7.

$$\begin{aligned} \boldsymbol{\mu}_{t-\frac{1}{2}} &= \boldsymbol{\mu}_{t-1} - \frac{\beta_t}{2}\Sigma_{t-1} g_t^f \\ g_t^f &= \partial f(\boldsymbol{\mu}) \end{aligned} \tag{7}$$

In the above update equation, $\frac{\beta_t}{2}$ is the common learning rate. Σ_{t-1} is the matrix to apply different learning rates to different coordinates.

The second step is a projection step with the smooth regularization penalty.

$$\begin{aligned} \tilde{\lambda} &= \frac{\beta_t}{2}\lambda \\ \boldsymbol{\mu}_t &= \arg\min_{\mu} \frac{1}{2}\|\boldsymbol{\mu} - \mu_{t-\frac{1}{2}}\|^2 + \frac{\beta_t}{2}\lambda|\Sigma_{t-1}\mu| \end{aligned} \tag{8}$$

The final updating rule goes to:

$$\mu_{t,j} = sign(\mu_{t-1,j} - \frac{\beta_t}{2}\Sigma_{t-1,jj} g_{t-1,j}^f)[|\mu_{t-1,j} - \frac{\beta_t}{2}\Sigma_{t-1,jj} g_{t-1,j}^f| - \frac{\beta_t}{2}\lambda\Sigma_{jj}] \tag{9}$$

4 Experimental results

4.1 test error rate vs sparsity

4.2 convergence rate vs sparsity

4.3 training time comparison

5 Reference