Exercise 4: Better online learning **Cenying(Tracy) Yang**
SDS 385 Due Date: 8/29/2016
Prof. James Scott

Working on Real Big Data

Regularization

Since the number of features is far larger than the number of observation, to prevent overfitting, regularization restricts the freedom of a model's parameters, penalizing their distance from some prior belief. Widely used regularizers penalize large weights with an objective function of the form

$$F(\boldsymbol{\beta}) = L(\boldsymbol{\beta}) + R(\boldsymbol{\beta}) \tag{1}$$

where $\boldsymbol{\beta}$ is the vector of coefficients of interested, and $L(\boldsymbol{\beta})$ is the loss function with respect to $\boldsymbol{\beta}$. Many commonly used regularizers $R(\boldsymbol{\beta})$ are of the form $\lambda||\boldsymbol{\beta}||$ where λ determines the strength of regularization and the ℓ_0, ℓ_1, ℓ_2^2, or ℓ_∞ are common choices for the penalty function.

Lazy Update

For a particular observation, the majority of feature value is zero. To address sparsity, lazy update is applied. The idea is to update a coefficient only when the feature is nonzero. The essence of the approach is given as follows.

Algorithm 1 Lazy Updates

Require: $\boldsymbol{\psi} \in \mathbb{R}^d$
  for $t \in 1, ..., T$ **do**
    Sample x_i randomly from the training set
    for j s.t. $x_{ij} \neq 0$ **do**
      $w_j \leftarrow Lazy(w_j, t, \psi_j)$
      $\psi_j \leftarrow t$
    end for
    $\boldsymbol{w} \leftarrow \boldsymbol{w} - \nabla F_i(\boldsymbol{w})$
  end for

We keep an array $\psi \in R^d$ in which each ψ_j stores the index of the last iteration at which the value of weight j was updated. When processing observation $\boldsymbol{x_i}$ at iteration k, we iterate through its nonzero features x_{ij}. For each such nonzero feature, we lazily apply the $k - \psi_j$ delayed updates collectively. Using the updated weights, we compute the prediction $\hat{y^{(k)}}$ with the fully updated relevant parameters from $\boldsymbol{\beta^{(k)}}$. We then compute the gradient and update these parameters. For **Ridge SGD Sparse Weight Update**, we have,

$$\beta_t^i = \beta_{t-n}^i (1 - \alpha\lambda)^n \tag{2}$$

where t is the current iteration, n is the collective steps we need to update, which is the difference between t and the last time β^i was updated, α is learning rate, and λ is the strength of regularization. For **Lasso SGD Sparse Weight Update**, we have

$$\beta_t^i = \beta_{t-n}^i \max\left(0, 1 - \frac{n\alpha\lambda}{|\beta_{t-n}^i|}\right) \tag{3}$$