Assignment 4 (15%): Training Classifiers via SGD. (Due on 16:59:59, Nov. 16, 2016)

Please implement two algorithms which are tailored to large-scale classification. Given the datasets with a large number of examples, we will train classifiers via stochastic gradient descent (SGD) in order to make the training phase more efficient.

Dataset

1. Data: We adopt two datasets which come from here. Each dataset contains both training set and testing set. (Please use editors like Sublime or Notepad++ to open them.)
   a. Dataset 1 (Training) Download, Dataset 1 (Testing) Download
   b. Dataset 2 (Training) Download, Dataset 2 (Testing) Download
2. Data Description and Format: Dataset 1 (Training) contains 32561 rows and 124 columns, and dataset 1 (Testing) contains 16281 rows and 124 columns; Dataset 2 (Training) contains 290507 rows and 54 columns, and dataset 2 (Testing) contains 290505 rows and 54 columns. Each row represents an example. The last column represents the label of the corresponding example, and the remaining columns represent the features of the corresponding example. For each dataset, label $\mathcal{Y} = \{-1, +1\}$.

Noted: The datasets maybe not be scaled, you should do it by yourself if necessary.

Task Description

Please accomplish the following two tasks:

Task 1: Binary classification with the *log-likelihood maximum logistic regression* with ℓ_1-norm on dataset 1 and 2.
Its objective can be formulated as ($\{x_i, y_i\}_{i=1}^N$ is the given dataset, each instance $x_i \in \mathcal{R}^d$ has a binary label $y_i \in \{-1, 1\}$; the goal is to learn a classifier $\beta \in \mathcal{R}^d$ from the data):

$$\min_{\beta} \frac{1}{N} \sum_{i=1}^{N} \{\log(1 + \exp^{-y_i \beta^T x_i})\} + \lambda ||\beta||_1$$

1. Training phase (on the training set): When you implement the SGD algorithm, you should take care of these points below:
   a. Please consider the bias term b in the optimization (we set $b = 0$ above for simplicity).
   b. The ℓ_1-norm is not smooth, so you should calculate the sub-gradient of it.
   c. The parameter $\lambda \geq 0$ serves as a balance weights. You can set it as a fix value or tune it on the training set by cross-validation.
2. Testing phase (on the testing set): Please test the classifier you have trained and record the test error in the testing set, i.e., comparing the label that the classifier predicts with the true label and recording the error rate.

Task 2: Binary classification with the *ridge regression* on dataset 1 and 2.
Its objective can be formulated as:

$$\min_{\beta} \frac{1}{N} \sum_{i=1}^{N} (y_i - \beta^T x_i)^2 + \lambda ||\beta||_2^2$$

The report should contain figures and tables that show the change of the test error and the objective function w.r.t. the number of iterations t for each loss function on each dataset. Specifically,

1. For the figures: figures are used to show the change of the objective function and training/testing errors clearly (please plot training and testing error in one graph). Thus, the x-axis of the figure is the number of iterations t in the training phase, and y-axis is the training/testing error using the classifier trained at a particular step. In addition, the y-axis can also be the objective function. By monitoring the change of the objective, the optimization procedure can be stopped when there is no obvious variation of it (You can set the total iteration number T by yourself). How to show the change well is also determined by yourself. For example, for each dataset and each loss function, you can plot the test error when $t = 0.01T, 0.02T, 0.03T, \ldots, 0.99T, T$ ($T = k * m$ and m is the number of examples in the training set and k is constant).
2. For the table: the numerical values in above figures.
   The report should contain the pseudo-codes of two SGD progress, the detailed experimental settings and the final results (figures and tables).

Further readings

Classification methods learn classifiers with labels. Most classification algorithms can be formulated as an optimization problem. By optimizing the objective function with data, the classifier can be trained. Stochastic optimization is useful when dealing with large-sclae datasets, which reduce the burden of gradient computation. The paper Pegasos: Primal Estimated sub-GrAdient SOlver for SVM uses a stochastic way to solve support vector machine, and it also produces some commonly used gradient computation examples. Some details of SGD method can be found in the Pegasos paper. The paper Stochastic Gradient Tricks gives many useful implementation tricks about SGD, which can be also very helpful. The liblinear package is a famous libear SVM implementation. It has a sparse logistic regression option (solved with batch optimization strategy). For instance, in matlab, you can use “model = train(label, sparse(train_data), ’-s 6 -c 1 -B 1 -q’)” to get the model. Ridge regression also has a closed form solution. We recommend you to compare your test error with the bath solution.

Reminders about Submission

1. Submission Deadline: **2016-11-16 16:59:59**.
2. Before submitting your assignment, please read **Submission Requirement and Description** section above carefully and obey it.
3. For assignment 3, please pack your **report**, **code** and **ReadMe.txt** into a zip file named with your student ID, e.g., MG1633001.zip.