CSC 576 - 2016 Fall: Homework 5 & Homework 6

Submission before the class on Nov. 24

Requirements

LaTeX generated Homework is preferred. *One additional point can be obtained if your homework is created from LaTeX*. Due to the request from some students, the homework is posted online right now, but would be updated probably every week until it is formally released. (*), (**), or (***) indicates the difficulty of each question. Please submit your homework *before* our class on the date above. Any late submission would not be accepted no matter what reasons.

Please indicate the names of your classmates if you discuss any question with them or ask for help from them.

(**) Question 1: 2 points

$f(\cdot)$ is a convex function. Prove that for any constant c,

$$\{x \mid f(x) \leq c\}$$

defines a convex set.

(**) Question 2: 3 points

Prove the follwing two statements are equavilent:

- the function $f(x)$ is convex;
- the epi graph of f is convex.

(***) Question 3: 5 point

Prove the following functions to be convex or nonconvex:

- $f(x) = x \log x$ for all x in $\mathbb{R}^{++}$;
- $f(x, y) = x^4/y$ for all y in $\mathbb{R}^{++}$;
- $f(x) = \| \max(0, Ax - b)\|^2$;
- $f(Z) = \sup_{\|x\|=1, \|y\|_\infty=1} x^\top Z y$;
- $f(x) = \log \sum_{i=1}^n \exp(x_i)$.

(***) Question 4: 4 points

Given a function $f(x, y)$ which is jointly convex in terms of x and y. Prove the following function is convex as well

$$g(x) := \min_y f(x, y).$$

(**) Question 5: 12 points

This is a programing homework. Consider the LASSO formulation we discussed in our class for many times:

$$\text{(LASSO)} \quad \min_x \quad f(x) := \frac{1}{2}\|Ax - b\|^2 + \lambda\|x\|_1,$$

where $\lambda > 0$ is a predefined parameter, $A \in \mathbb{R}^{n\times p}$ is the data matrix, and $b \in \mathbb{R}^n$ is the observation vector.

Follow the procedure blow to generate the data:

- Generate a sparse vector x^* with 20 nonzero elements (the value of nonzero elements can be generated from Gaussian Distribution $\mathcal{N}(0, 10)$);
- Generate $A \in \mathbb{R}^{n\times p}$: all elements of A follow i.i.d. Gaussian distribution $\mathcal{N}(0, 1)$;
- Generate $b \in \mathbb{R}^n$: $b = Ax^* + \epsilon$ where all elements ϵ_i's follow i.i.d. Gaussian distribution $\mathcal{N}(0, 1)$;
- Set $\lambda = \sqrt{2n \log p}$.

Please implement the following four algorithms to solve this problem:

- (Proximal) Gradient Descent (you can use line search to decide the step length or constant step length $\gamma = 1/\|A^T A\|$);
- Accelerated (Proximal) Gradient Descent (you can use line search to decide the step length or constant step length $\gamma = 1/\|A^T A\|$);
- Coordinate Descent (exactly optimize each coordinate);
- ADMM.

Set the initial point as $x_0 = 0$ in all four algorithms and plot their objective function values at each iteration. Note that one iteration is defined as updating all coordinates in x once, which means that the complexity of four algorithms per iteration are comparable. Compare and comment the efficiency for four algorithms.

Submit your runnable code to blackboard and make sure that your result is reproducible.

(**) Question 6: 10 points

In this experiment, you need to use linear SVM to do classification task on Gisette dataset, which is a dataset with binary labels. For more detailed information of this dataset, see "https://archive.ics.uci.edu/ml/datasets/Gisette". Here we use the validation set as test data, the data is provided on the course website, the training data and test data (labels) are in the file named "train.data(.labels)" and "test.data(.labels)". Each row corresponds to one sample, and each sample have 5000 features.

In this experiment, you need to implement the linear Support Vector Machine (SVM) with L_1 and L_2 regularization. You are **NOT** allowed to use existing machine learning libraries (e.g. LIBSVM) which have SVM implementations (but you can use libraries which provides other basic operations e.g. numpy). You need use the optimization methods which you learn from this class (e.g. (proximal) gradient descent, (proximal) stochastic gradient descent, (proximal) stochastic coordinate descent, etc.) to directly implement a linear SVM solver.

You can use the following formulations:

$$\min_{\mathbf{w}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C_2 \sum_i \max(0, 1 - \mathbf{y}_i X_{i,:}\mathbf{w}) \qquad [L_2\text{-regularized SVM}]$$

$$\min_{\mathbf{w}} \quad \|\mathbf{w}\|_1 + C_1 \sum_i \max(0, 1 - \mathbf{y}_i X_{i,:}\mathbf{w}) \qquad [L_1\text{-regularized SVM}]$$

where C_1 and C_2 are hyper parameters.

Besides implement the experiment, you need to provide an experiment report, which needs to contain:

- Your updating equation showing how to obtain $\mathbf{w}_{k+1}$ from $\mathbf{w}_k$ for both algorithms.
- The specific method, the parameter (e.g. step length, loss penalty parameter C), and any other information which is needed to reproduce your experiment.
- Figures to show the training/test error (percentage of correctly classified samples), and iteration number. (error is shown in y-axis, and iteration number is shown in x-axis)
- Compare the final training/test error for both L_1-regularized SVM and L_2-regularized SVM.
- How many non-zero elements in the weights vector $\mathbf{w}$ for L_1-regularized SVM? What about the L_2-regularized SVM? (This actually means how many features you select from the 5000 features.)