SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

50.570 Machine Learning, Fall 2017

Assignment 2

Last update: Monday 9th October, 2017 00:06

Grading Policy and Due Date

- You are required to submit: 1) a report that summarizes your experimental results and findings, based on each of the following question asked; 2) your implementation (source code) of the algorithms.
- You are free to choose any programming language you prefer.
- Submit your assignment report and code to eDimension.
- This assignment is an individual assignment. Discussions amongst yourselves are allowed and encouraged, but you should write your own code and report.
- Submit your assignment to the eDimension by 11:59 PM on Sunday 22 October 2017. Late submissions will be heavily penalized (20% deduction per day).

Task 1: SVM Implementation

We would like to implement SVM using quadratic programming discussed in class. Please first download the optimizer by following this link: `http://tinyurl.com/50-570-optimizer`. You may need to obtain a free Academic License in order to use the optimizer. Please also download `a2-data.zip` which contains the data needed for this task. There are three datasets (data A, data B and data C). Each dataset consists of a training and a test set. You are required to report results on the three datasets separately.

1. (10 pts) Implement a simple linear SVM that deals with the case where the data is slightly not linearly separable (*i.e.*, there are slack variables) in the primal form and in the dual form. Write a prediction function that can be used to predict an input point. Run your code on the training data to build the model, with $C = 1.0$ (note $C = 1/\lambda$), and evaluate your model's performance on the training and test data. Report your results and findings clearly using tables.

   Hint: You should be able to verify the correctness of your implementations by comparing the results obtained from the two implementations.

2. (10 pts) Discuss the behavior of the performance on the training and test set as we change the value of C (or $1/\lambda$). What happens to the margin as C increases? What happens to the number of support vectors as C increases? Use graphs and tables to show such results clearly.

3. (10 pts) Typically, a different value of C corresponds to a different model for classification. What is a reasonable way to select C? Do you believe we can optimize the value of C by maximizing the margin on the training set? If so, give your argument and provide empirical support for your argument. If not, explain why not, and propose and implement a different method to select a good value of C so as to obtain a good performance on the test set. Report your performance on the training and test set with the C selected by your method.

4. (10 pts) Extend your implementation to support kernels. (Please note that only minimal modifications will be required.) Try to use the Gaussian kernel and another kernel of your choice and perform experiments on the three data sets. Use tables to clearly present your results.

5. (10 pts) In class we have discussed that the sum of two kernels is a kernel and the product of two kernels is again a kernel.

   Formally prove that the product of two kernels is a kernel. In other words, if $K_1(x, x')$ and $K_2(x, x')$ are both kernel functions, then $K_1(x, x')K_2(x, x')$ is again a kernel.

Task 2: Logistic Regression

Now, let us implement the logistic regression model. Note that the logistic regression model minimizes the following objective function:

$$\mathcal{E}_n = \sum_{i=1}^{n} \log\left(1 + \exp(-y^i(\theta \cdot \mathbf{x}^i + \theta_0))\right)$$

We can also add a regularization term to it, leading to:

$$\mathcal{E}_n = \sum_{i=1}^{n} \log\left(1 + \exp(-y^i(\bar{\theta} \cdot \mathbf{x}^i + \theta_0))\right) + \lambda||\theta||^2 \tag{1}$$

1. (10 pts) Implement the regularized logistic regression using stochastic gradient descent. Perform experiments using the same data used in the previous question and compare the performance with that of SVM in the previous question. Try setting different values for λ (*e.g.*, 0.001, 0.01, 0.1, 1, 10, 100, 1000) and report the different results obtained. Also report the learning rate you used and the final objective value obtained when the algorithm converges.

2. (10 pts) Try using different learning rates in your implementation of stochastic gradient descent. Did you see different final objective value when a different learning rate is used? Clearly and concisely explain why.

3. (10 pts) Let us now consider the following new objective function:

$$\mathcal{E}_n = \sum_{i=1}^{n} \log\left(1 + \exp(y^i(\bar{\theta} \cdot \mathbf{x}^i + \theta_0))\right) + \lambda||\theta||^2 \tag{2}$$

   Modify your implementation and clearly report the performance on the three datasets. Compare the performance with your previous implementation that optimizes (1) (under the same λ values). Try to explain your observations.