Homework 7

Due May 29, 2017 by 11:59pm

Instructions: Upload your answers to the questions below to Canvas. Submit the answers to the questions in a PDF file and your code in a (single) separate file. Be sure to comment your code to indicate which lines of your code correspond to which question part.

There is 1 reading assignment and 2 exercises in this homework. The purpose of Exercise 2 is to prompt you to make progress on the data competition.

Note that the reading assignment and Exercise 2 are not graded as part of this homework. You may consider Exercise 2 as a milestone for your data competition project, which you can add already to your data competition final report. Still, we recommend you to submit your work on Exercise 2 with this homework, so that we can adapt the labs to fit your needs in order to make progress on the data competition.

"Reading" Assignment

Read and play with scikit-learn's function `SVC` using the example:

`http://scikit-learn.org/stable/auto_examples/svm/plot_iris.html#sphx-glr-auto-examples-svm-plot-iris-py`

Visualize the results as you change the kernel and/or vary the kernel parameters.

1 Exercise 1

In this exercise, you will implement in **Python** a first version of your own linear support vector machine with the squared hinge loss.

Recall from the lectures that the linear support vector machine with the squared hinge loss writes as

$$\min_{\beta \in \mathbb{R}^d} F(\beta) := \frac{1}{n} \sum_{i=1}^{n} \left(\max(0, 1 - y_i x_i^T \beta) \right)^2 + \lambda \|\beta\|_2^2 \,. \tag{1}$$

You know now by heart the fast gradient algorithm, so no need to recall it here.

- Compute the gradient $\nabla F(\beta)$ of F.
- Consider the Spam dataset from *The Elements of Statistical Learning*. Standardize the data, if you have not done so already.

- Write a function *mylinearsvm* that implements the fast gradient algorithm to train the linear support vector machine with the squared hinge loss. The function takes as input the initial step-size value for the backtracking rule and a maximum number of iterations.

- Train your linear support vector machine with the squared hinge loss on the the Spam dataset for the $\lambda = 1$. Report your misclassification error for this value of λ.

- Run cross-validation to find the optimal value of λ. Report your misclassification error for that value of λ.

2 Data Competition Project

In this exercise, you are going to train support vector machines (SVMs) using **scikit-learn** and the data competition project dataset. You will consider here *all classes* in the dataset. You may work on this exercise on your own computer first. Note, however, that you **need** AWS to run the experiments for the last two parts of this exercise.

Warm-up

- In a one-vs-one fashion, for each pairs of classes, train a linear SVM classifier using scikit-learn's function `LinearSVC`, with the default value for the regularization parameter. Compute the *multi-class misclassification error* obtained using these classifiers trained in a one-vs-one fashion.

- In a one-vs-rest fashion, for each class, train a linear SVM classifier using scikit-learn's function `LinearSVC`, with the default value for λ_c. Compute the multi-class misclassification error obtained using these classifiers trained in a one-vs-rest fashion.

- Using the option `multi_class='crammer_singer'` in scikitlearn's function `LinearSVC`, train a multi-class linear SVM classifier using the default value for the regularization parameter. Compute the multi-class misclassification error obtained using this multi-class linear SVM classifier.

Linear SVMs for multi-class classification

- Redo all questions above now tuning the regularization parameters using cross-validation.

Kernel SVMs for multi-class classification

- Redo all questions above now using the polynomial kernel of order 2 (and tuning the regularization parameters using cross-validation).