CSE 5526 – Autumn 2014

Introduction to Neural Networks

Programming Assignment 3: SVM

Due Thursday, November 13

Grader: Yuzhou Liu
Email: liu.2376@osu.edu
Office: Dreese Labs 578
Office hours: Monday & Wednesday 3 - 4pm

In this project, we will use LIBSVM, a popular SVM toolkit, to analyze the data from the following paper:

Uzilov, A. V., Keegan, J. M., & Mathews, D. H. (2006). Detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change. BMC bioinformatics, 7(1), 173.

Non-coding RNA (ncRNA) sequences serve a multitude of roles in a cell, many of which are still to be discovered. While it is difficult to detect novel ncRNAs in biochemical screening, some studies have shown that computational methods can accurately detect ncRNAs from genetic sequence data. The approach that we will adopt treats ncRNA detection as a classification problem. Each data point is a pair of homologous RNA sequences from two organisms. The prediction task is to decide whether both of them are ncRNA or both are not. The positive instances in this task are known ncRNA sequence pairs, and the negative instances are derived from the positive instances by shuffling both sequences. To perform the classification, an 8-dimensional feature vector is extracted from each pair:

1. A feature value computed by the “Dynalign algorithm” between the two sequences
2. The length of shorter sequence
3. 'A' frequencies of sequence 1
4. 'U' frequencies of sequence 1
5. 'C' frequencies of sequence 1
6. 'A' frequencies of sequence 2
7. 'U' frequencies of sequence 2
8. 'C' frequencies of sequence 2

You should use LIBSVM for the project, which is a popular open-source SVM toolbox that can be called from many programing languages, such as C/C++, JAVA, and MATLAB. Download LIBSVM from this webpage: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

The training and test data can be downloaded from Carmen in the “Assignments” topic. They are saved in the LIBSVM format. Use the wrapper functions provided in the package for your language of choice to read and write the files. For example, for MATLAB users, use the `libsvmread` and `libsvmwrite` functions. Note that there is a guide for LIBSVM: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf and the README file in your downloaded package contains very helpful information for using LIBSVM.

Perform the following experiments:

1. Classification using linear SVMs. Perform the following two steps:
   a) Use 5-fold cross validation to choose the best C parameter. To do so, divide the training set into 5 subsets of equal size. Each subset in turn is used to validate the classifier trained on the remaining 4 subsets. So you have 5 trained SVMs and 5 validation subsets. The cross-validation accuracy is the average accuracy over all of the data points you have validated on. Note that you should NOT use the built-in cross validation option in LIBSVM, but should instead write your own cross-validation code and code to search for the best C. Try the values in the following exponential steps for C: (2^{-4}, 2^{-3},…, 2^{7}, 2^{8}). Plot your cross validation accuracy as a function of C (use MATLAB's `semilogx` function instead of `plot` so that these values of C appear evenly spaced).
   b) Use the whole training set to train an SVM with the best *C* value you have found in the previous step. Use the trained SVM to classify the test set and report the (scalar) classification accuracy.
2. Classification using RBF kernel SVM: $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\alpha\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2)$.
   a) Use 5-fold cross-validation to choose the best C and α parameters. Use the same division of data points into subsets that you used for step 1a above. Do this for all pairs of C and α values from the set (2^{-4}, 2^{-3},…, 2^{7}, 2^{8}). This should produce a 13x13 matrix of accuracies, where the accuracy in row i of the matrix corresponds to the ith value of C and the accuracy in row j of the matrix corresponds to the jth value of α. Plot these accuracies as a heat map (e.g., using matlab's `imagesc` command) with the axes clearly labeled or report them as a matrix.
   b) Use the whole training set to train an SVM with the best C and α values that you found in the previous step. Use the trained SVM to classify the test set and report the (scalar) classification accuracy. Compare it to the accuracy for the linear SVM.

Turn in the following to the project's dropbox on Carmen:

1. A 1-2 page summary report, including the requested plots. The report should explain the question that you are trying to answer, what you did to answer it, and the results that you found. It should discuss whether those results are surprising or not given what we have learned in class. If they are surprising, it should propose some plausible explanations.
2. Your source code
3. A script or executable that can be run with no arguments to generate the plots in your report and any other evidence that your implementation is working properly.
4. A README file identifying and explaining the script/executable from 3.