CS-584 – Assignment 4 (5%)

Support Vector Machines

Due by: April 14, 2016

Assignment Specifications

In this assignment you will implement Support Vector Machine algorithms. You have to use two or more external datasets. One of the datasets needs to have ordinal and/or nominal features and must have missing features. Links to datasets are available on the course web-page (e.g. the UCI Machine Learning Repository http://archive.ics.uci.edu/ml/). It is essential that you evaluate the performance of each algorithm you implement and the effects of varying different parameters on the performance of the learning algorithm. Use cross validation to test performance. The grade for this assignment will be based in part on the performance of your implementation and on the thoroughness of your evaluation. Make sure to explain the results you obtain and do not unnecessarily repeat similar results. The code you write should be modular and well documented. The implementation needs to be in Python. You need to implement the SVM yourself. Use the scikit-learn version to verify your implementation.

1. Generate a small dataset of 2D feature vectors of two classes such that the classes are linearly separable. Similarly, generate an additional set with examples that are not separable.

2. Implement a linear SVM algorithm with hard margins and apply it to the separable dataset you generated. Plot the data points and mark the support vectors you identified. Apply the algorithm you implemented to the non-separable dataset and observe the results. Make sure to normalize the data if necessary (in this and subsequent algorithms you implement).

3. The primal objective function of soft margins SVM is given by:

$$L_P = \frac{1}{2}||W||^2 + c\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i(y^{(i)}(W^T x_i + w_0) - 1 + \xi_i) - \sum_{i=1}^{m}\beta_i\xi_i \quad (1)$$

   Using the minimization equations $\frac{\partial L_P}{\partial w_i} = 0$, $\frac{\partial L_P}{\partial w_0} = 0$, and $\frac{\partial L_P}{\partial \xi_i} = 0$ develop the expression of the dual L_D that has to be maximized. You need to show and explain the development in addition to showing the final result.

4. Implement a linear SVM algorithm with soft margins and apply it to the datasets you generated. Plot the data points and mark the support vectors you identified.

5. Implement a kernel-based SVM algorithm using a polynomial and Gaussian kernel functions. Apply your implementation to the datasets you generated. Plot the support vectors you obtain. Test the algorithm on two additional external datasets and report performance. Test and explain the effect of modifying different parameters of the algorithm.

6. Test what happens when one of the classes has substantially more examples. Implement and test a solution for this case.

Please follow the submission instruction of assignment 1.