Support Vector Machines

Due date: Wednesday, May 17, in-class

Assignment

The assignment is fairly straightforward.

1. Search UCI Machine Learning Repository, Kaggle, or any other dataset repository for datasets you may like.

2. Select three datasets on which to perform Support Vector Machine classification.

   You can select datasets with any types of independent variables (perform the replacement of categorical variables with dummy variables if you have categorical variables in your data). Your dependent variable must be a categorical class variable. Ideally, your dependent variable is a binary class variable. If you select a problem where the class variable is non-binary, you need to either combine class labels into two classes (i.e., simplify your classification problem), or create "one vs. the rest" binary classification problems for each class label.

   At least one of your selected problems shall have only two numeric independent variables (so that you could visualize it). Other problems can (and should) have more independent variables.

3. Implement (from scratch) the gradient descent solution for the Linear Kernel Support Vector Machine optimization problem, and apply it to each of your classification problems.

4. Compute the accuracy of your SVM model for each of the problems. Report the accuracy, and also, report the support vectors discovered by your model.

5. See if you can use the kernel trick to improve your classification accuracy.

6. For the two-dimensional case, visualize all your results. For other cases, simply report them.

Use of Jupyter Notebooks. In general, there is no restriction on the use of programming language/envioronment, etc. However, it may be easiest for you to meet the requirements of the assignment by creating and demonstrating Jupyter Notebooks.

Use of existing tools. You must perform the computations that constitute the gradient descent solution for SVMs yourselves. You can use standard SVM classifiers from existing machine learning and statistical packages as the means of testing your code. You can use utility functionality from existing machine learning and statistical packages. You can use numpy for matrix manipulation.

Deliverables

Have your code in a demonstratable form starting May 17, 2017. You will be asked to demo your work in class. You should be able to give a short (5 mins or so) demo describing the independent and dependent variables you used, and showing the resulting regression line/plane and the regression equations.

Have your code in a ready-to-submit form. Submission instructions will be provided separately.