Assignment 06

Machine Learning, Summer term 2018
Norman Hendrich, Marc Bestmann, Philipp Ruppel
May 14, 2018

Solutions due by May 27

Assignment 06.1 (SVM in scikit-learn, 1+2 points)

For this exercise, please familiarize youself with the Support Vector Machine algorithms provided in the Scikit-Learn Libary.

a. Install Sklearn on your computer and see `http://scikit-learn.org/stable/modules/svm.html` for documentation and examples of the available variants of the SVM algorithm.

b. Read the documentation for `sklearn.svm.SVC`, the kernel-based SVM classifier: `http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`

   How do you configure and train the SVM? How do you classify new test points? How do you access the support vectors after training? How do you configure the SVM for multi-class classification problems?

Assignment 06.2 (SVM cancer detection, 2+3+1 points)

In this exercise you train a (soft margin) SVM that classifies cancers as either benign (-1) or malignant ($+1$) depending on the characteristics of sample biopsies. For this exercise, use the SVC module from `sklearn`.

Load the patients data from `cancer-data.mat` (available on the course webpage). For every patient, 9 attributes are measured: 1. clump thickness, 2. uniformity of cell size, 3. uniformity of cell shape, 4. marginal adhesion, 5. single epthelia cell size, 6. bare nucle, 7. bland comatin, 8. normal nucleoli, 9. mitoses.

a. For $C \in \{0.01, 0.1, 0.5, 1, 5, 10, 50\}$, plot the train and test error with respect to the 0-1-loss as a function of C for a linear SVM. What is the effect of choosing a large C on the training error? Does this effect coincide with what you are expecting?

b. Now, try out different kernel functions. Find optimal kernel parameters and C by cross-validation. Which SVM kernel performs best on the test data?

c. Exchange the train and test sets (train with 'cancerInput_test' and test with 'cancerInput_train'). What bevaviour do you observe now?

Assignment 06.3 (Computational complexity, 1+1+1 points)

a. Estimate the prediction running time for linear least-squares, d kNN regression (computational complexity) as a function of d (input dimensions), n (number of training points) and k (number of nearest neighbors).

b. What is the computational complexity of predicting a new data point for a SVM? (m support vectors after training).

c. How much information do you need to store for predicting with each of these methods (space complexity)?

d. For a specific example, consider a one-against-the-rest classifier for the full USPS dataset ($d = 256$, $n = 10000$), assume $k = 10$ for kNN and $m = 1000$ support vectors for the SVM classifier. How many operations and how much memory is needed?