Lab Exercise 5

SVM classifier with WEKA

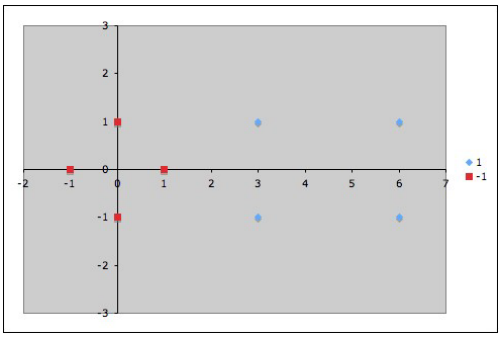
Exercise 1: Basic manual classification using SVM

Suppose we are given the following positively labeled data points in R^2:

$$\left\{\begin{pmatrix}3\\1\end{pmatrix},\begin{pmatrix}3\\-1\end{pmatrix},\begin{pmatrix}6\\1\end{pmatrix},\begin{pmatrix}6\\-1\end{pmatrix}\right\}$$

and the following negatively labeled data points in R^2 (see figure):

$$\left\{\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}0\\-1\end{pmatrix},\begin{pmatrix}-1\\0\end{pmatrix}\right\}$$



Discover a simple SVM that accurately discriminates the two classes. Since the data is linearly separable, we can use a linear SVM (that is, one whose mapping function Φ() is the identity function)

Using the obtained SVM classification function classify point x = (4, 5).

Exercise 2: Spam filtering with WEKA and SVM classifier

In this exercise we re-examine the spam filtering problem from Lab 4. This time, we train a a linear Support Vector Machine for the spam or non-spam classification task.

1. Start up Weka, select the Explorer interface and load the preprocessed *Spambase* data set from Lab 4, where all attributes are converted to Boolean and randomize the instances. If you did not save this in Lab 4 Ex. 2 as instructed, then go back and perform preprocessing task.
2. Now it's time to train our classifiers. The task is to classify e-mails as spam or non-spam and we evaluate the performance of Support Vector Machines on this task and compare to the performance of Naïve Bayes from Lab 4.
3. Go to the the **Classify tab**, select **Choose > functions > SMO** (SMO stands for Sequential Minimal Optimization, which is an algorithm for training SVMs). Use the default parameters and click **Start**. This will train a linear SVM. Select the percentage split and set it to 10%. This is done in order to save us waiting while Weka works hard on a large data set.
4. Click Start to train the model. Examine the Classifier output frame to view information for the model you've just trained and try to answer the following questions:
    - What is the percent of correctly classified instances? How does it compare to the result from Naïve Bayes?
    - How do the regression coefficients for class 1 relate to the ones for class 0?
    - What are the coefficients for the attributes [word_freq_hp_binarized] and [char_freq_$_binarized]? Generally, we would expect the string $ to appear in spam, and the string hp to appear in non-spam e-mails. Do the regression coefficients make sense given that class 1 is spam?