Perceptron vs SVM analysis

Tommi Jaakkola, course materials for Machine Learning, Fall 2006.
MIT OpenCourseWare (http://ocw.mit.edu/)

1 Percepton

Implement a Perceptron classifier. For this problem, we have provided you two custom-created datasets. The dimension d of both the datasets is 2, for ease of plotting and visualization.

(a) Load datasets using the **load_p1_a** and **load_p1_b** scripts and on each of them train a Perceptron classifier until it makes no errors on the training data.

(b) What is the angle between *theta* and the vector $(1, 0)^T$ for each dataset?

(c) What is the number of updates k_a and k_b required before the Perceptron algorithm converges?

(d) Compute the geometric margins, $\gamma^a_{geom_-}$, $\gamma^a_{geom_+}$ and $\gamma^b_{geom_-}$, $\gamma^b_{geom_+}$, of your classifiers with respect to their corresponding training datasets and their class. Recall that the distance of a point x_t from the line $\theta^T x = 0$ is $|\frac{\theta^T x_t}{\|\theta\|}|$.

(e) For each dataset, plot the data (as points in the $X - Y$ plane), along with the corresponding decision boundary that each Perceptron classifier computed. Your plots should clearly indicate the class of each point (e.g., by choosing different colors or symbols to mark the points from the two classes).

Solution

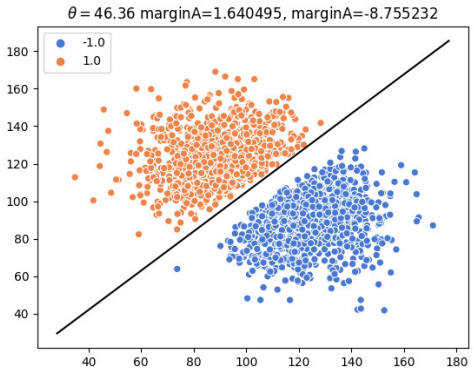

Figure 1: Dataset A

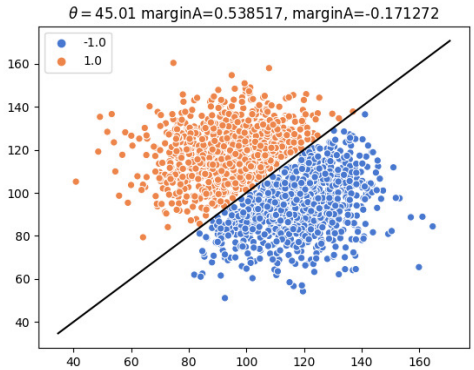

Figure 2: Dataset B

2 SVM

Implement an SVM classifier.

Hint: Use the built-in quadratic program solver $quadprog(H, f, A, b)$ which solves the quadratic program: $\min\{\frac{1}{2}x^T Hx + f^T x\}$ subject to the contraint $Ax \leq b$.

(a) Try the SVM on the two datasets from Problem 1. How different are the values of *theta* from the values the Perceptron achieved? To do this comparison, should you compute the difference between two vectors or something else?

(b) For the decision boundaries computed by SVM, compute the corresponding geometric margins (as in Problem 1(d)). How do the margins achieved using the SVM compare with those achieved by using the Perceptron?

Solution

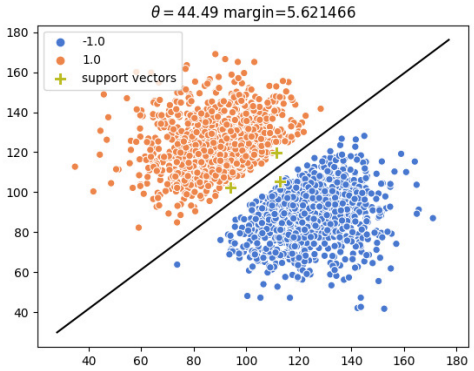

Figure 3: Dataset A

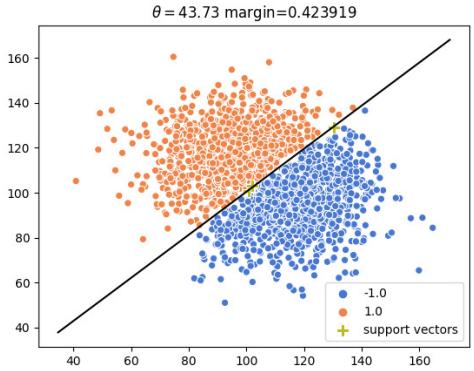

Figure 4: Dataset B

3 SVM light

Now that you are familiar with SVMs, download this highly optimized version called SVM_{light}: **http://svmlight.joachims.org/**

We will use this package to experiment with classifying images of handwritten digits. To simplify things, we will only be distinguishing between 0s and 1s.

(a) Train a plain-vanilla SVM (i.e, no regularization) on **train-01-images.svm** (which is already in SVM_{light}'s format). What is the training error, i.e., the fraction of examples in the training data that are misclassified? From the set of misclassified images, pick one. Attach of plot of it with your solution. Why do you think the SVM fails to classify it correctly during training? Now apply the SVM to the test set **test-01-image.svm**; what is the test error, i.e., the fraction of test data that is misclassified?

(b) Experiment with different values of the regularization term C (set using the -c flag). Start by guessing/estimating a range in which you think C should lie. Then choose the values of C (within that range) at which you will evaluate the performance of the SVM. You need not pick more than 10 such values, though you should feel free to pick as many (or as few!) as you want. For these values of C, plot the corresponding error. What value of C gives you the best training error on the dataset from part (a)? How does the test error

for this choice of C compare with the test error you computed in part (a)? Could you optimize C so that it leads to the best test error, rather than the training error? Should you?

(c) An adversary (we'll call him W) mislabeled 10% of the images in the original training set to produce **train-01-images-W.svm**. Using the same choices of C as in part (b), find the C that results in the best training error. What is the corresponding test error on **test-01-image.svm**?

Solution

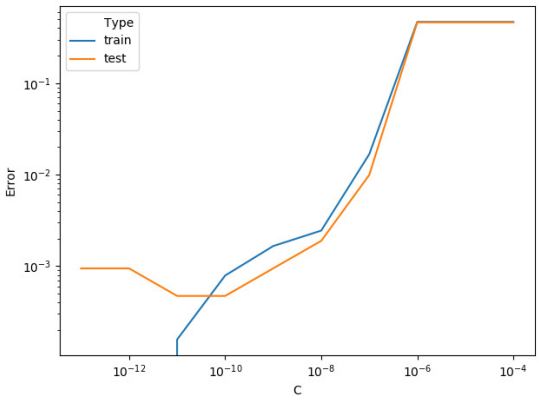

Figure 5: Original data

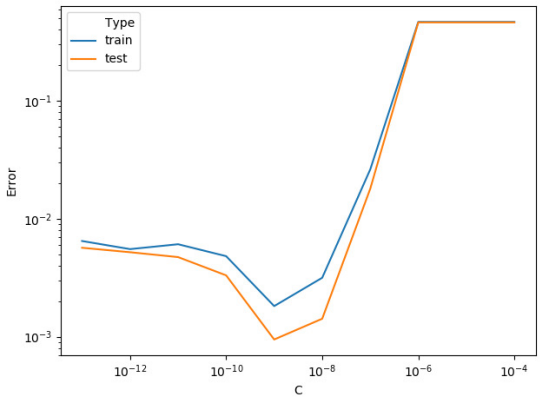

Figure 6: 10% misclassified data

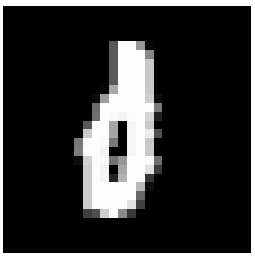

Figure 7: Example of misclassified instance

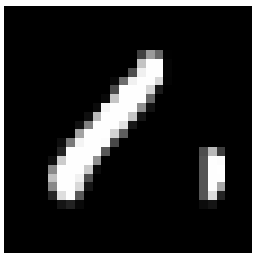

Figure 8: Example of misclassified instance