Applied Data Mining (UN3106)
Spring 2018
http://stat.columbia.edu/∼porbanz/UN3106S18.html

Peter Orbanz
porbanz@stat.columbia.edu
Phyllis Wan
pw2348@columbia.edu

Homework 3

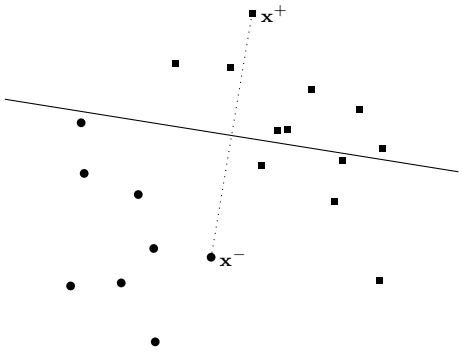
Due: 5 April 2018

Homework submission: We will collect your homework **at the beginning of class** on the due date. If you cannot attend class that day, you can leave your solution in Phyllis Wan's postbox in the Department of Statistics, 10th floor SSW, at any time before then.

We do not accept homework submitted late. There will be no exceptions.

Problem 1 (A primitive ensemble)

This problem considers a very simple ensemble classifier for a two-class problem, based on the idea of "randomly throwing out hyperplanes" discussed in class.



The basic idea is to define a "weak classifier" as follows:

- Choose two training data points $\mathbf{x}^-$ and $\mathbf{x}^+$, one in each class.
- Place an affine plane "in the middle" between the two, i.e. purpendicular to the connecting line. The connecting line is drawn in the figure as a dotted line, the affine plane as a solid line. In formulas, the plane is given by a normal vector $\mathbf{w}$ and an offset $c_{\mathbf{w}}$ computed as

$$\mathbf{w} := \frac{\mathbf{x}^+ - \mathbf{x}^-}{\|\mathbf{x}^+ - \mathbf{x}^-\|} \quad \text{and} \quad c_{\mathbf{w}} := \left\langle \mathbf{w}, \mathbf{x}^- + \frac{1}{2}(\mathbf{x}^+ - \mathbf{x}^-) \right\rangle$$

- Try out each of the two possible orientations of the plain; select the one with the smaller training error. That is, the weak classifier is defined as

$$f(\,.\,) = \mathrm{sgn}(\langle\,.\,, \mathbf{v}\rangle - c)$$

  where

$$\text{either } \mathbf{v} := \mathbf{w} \text{ or } \mathbf{v} := -\mathbf{w} \quad \text{and} \quad c := \left\langle \mathbf{v}, \mathbf{x}^- + \frac{1}{2}(\mathbf{x}^+ - \mathbf{x}^-) \right\rangle,$$

  depending on which choice achieves smaller error.

We combine many of these weak classifiers into a single ensemble classifier.

More precisely:

- Split the available data into to equally sized parts (training and test).
- Select m pairs of points $(\mathbf{x}_1^-, \mathbf{x}_1^+), \ldots, (\mathbf{x}_m^-, \mathbf{x}_m^+)$, where each $\mathbf{x}_i^-$ is drawn uniformly at random from class -1,and $\mathbf{x}_1^+$ uniformly from class $+1$. Sample with replacement, i.e. training data points can appear more than once.
- For each such pair $(\mathbf{x}_i^-, \mathbf{x}_i^+)$, compute the classifer f_i given by $(\mathbf{v}_i, c_i)$ as described above.
- The overall classifier g_m is defined as the majority vote

$$g_m(\mathbf{x}) = \text{sgn}\Big(\sum_{j=1}^{m} f_i(\mathbf{x})\Big) = \text{sgn}\Big(\sum_{j=1}^{m} \text{sgn}(\langle \mathbf{v}_i, \mathbf{x}\rangle - c_i)\Big).$$

We will try to understand how well this classifier performs depending on how many weak classifiers we include in the computation.

Questions:

1. Implement the classifier g_m above in R: Write a training function $[\mathtt{V}, \mathtt{C}] = \mathtt{train\_g}(\mathtt{m}, \mathtt{trainingData})$ that returns a $(m \times d)$-matrix $\mathbf{V}$ and a length m-vector $\mathbf{C}$ (d is the length of $\mathbf{x}$), where

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \cdots \\ \mathbf{v}_m \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{pmatrix} c_1 \\ c_2 \\ \cdots \\ c_m \end{pmatrix}.$$

2. Write a function $\mathtt{classify}(\mathtt{x}, \mathtt{V}, \mathtt{C})$ that applies the classifier to a new data point $\mathbf{x}$. This function should return ± 1.

We perform experiments on two data sets: The USPS digit data we have already used before, and another data set often used as an example, the Wisconsin Breast Cancer Data (available on the course page). For each data set:

3. Randomly and equally separate the data into a training and a test data set. (Subsetting the data does not perform well in the WBCD dataset. I tried using loops. The time is not that bad.)
4. For each $m = 1, 3, \ldots, 199$:
   (a) Train a classifier g_m.
   (b) Classify each point in the test set.
   (c) Compute the error rate on the test set.
5. Record all errors and plot the error rate as a function of m.

Optional:

6. You can also further increase the range of m to see what happens.