1 Statistical Classifiers and Distance Functions

Problem: SVM and decision trees don't know about symmetry and transitivity.

Solution: Massage your data so that more of the symmetry is apparent in the feature space.

Tricky ways of enforcing symmetry:

1. Concatenate feature vectors and add them twice. For instance, for datapoints x and y, $[x, y]$ is a single vector and $[y, x]$ another. Useful for SVM.

2. Mix it up: $[x+y, \text{pos}(x-y)]$ where the function $\text{pos}(x)$ ensures that the first nonzero element is positive. Useful for boosted decision trees.

2 Extra (Unlabeled) Data

Used only to help with the partitioning of the space with the Gaussian mixture models. Other people tried to add them when minimizing the loss in the boosting step. This was a bad idea since it made a classifier better if it partitioned unlabeled points well (even if it did so incorrectly).

3 Boosting the Weak Learners

Learning on $(x_1, y_1), \ldots, (x_n, y_n)$ where x_i are features and the y_i are ± 1 depending on whether x_i is in the particular class we're learning, or y_i denotes that this data point is unlabeled.

Let $D_1(i) = 1/n$ be the initial, uniform, distribution vector.

As many times as you want ($t = 1, \ldots$):

1. Given a distribution, use a Gaussian mixture model to partition the space using an EM algorithm.

2. Find a hypothesis function h_t which takes a data point and sends it to $[-1, 1]$ based on whether it's in the training class. Make sure that $correctness_t = \sum_i D_t(i) h_t(x_i) > 0$. If you can't find a hypothesis which does this, bail.

3. Define $strength_t = \frac{1}{2} \ln \left(\frac{1+correctness_t}{1-correctness_t} \right)$

4. Update the distribution

$$D_{t+1}(i) = \begin{cases} D_t(i) \exp(-correctness_t \; y_i \; h_t(x_i)) & y_i = \pm 1 \\ D_t(i) \exp(-correctness_t) & y_i \; unlabeled \end{cases}$$

5. Normalize the distribution vector D_{t+1} to sum to 1.

The end result is the function

$$f(x) = \sum strength_t \; h_t(x).$$

Note that this outputs something between $[0, 1]$ depending on how close we are to the training class.

4 Mahalanobis Distance

Say you have a collection of values with mean $\mu = (\mu_1, \mu_2, \ldots, \mu_p)^T$ and a covariance matrix Σ.

Then the distance of a vector $x = (x_1, x_2, \ldots, x_p)^T$ is given by

$$D(x) := \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

Note that $x^T M x = r$ is a ellipse.

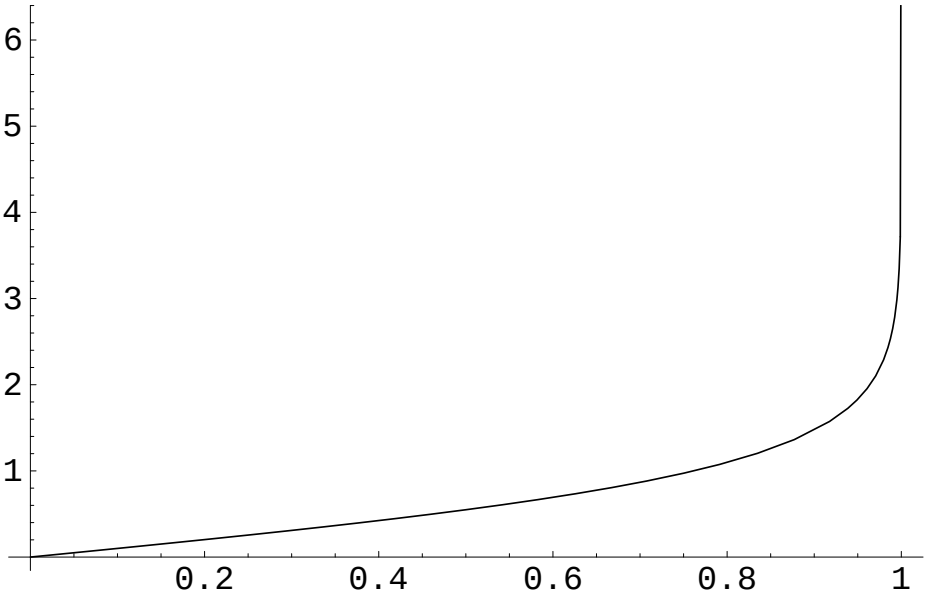

Figure 1: Plot of $\frac{1}{2}\ln\left(\frac{1+x}{1-x}\right)$