

CMPS 242 Second Homework, Fall 2006

2 Problems, due start of class Friday October 27

This homework is to be done in groups of 2-3. Each group member should completely understand the group's solutions and *must* acknowledge all sources of inspiration, techniques, and/or helpful ideas (web, people, books, etc.) other than the instructor and class text.

1. Construct a small data set with boolean features where the greedy (univariate) decision tree procedure using the impurity heuristic (Equation 9.8 in Alpaydin) fails to find a smallest decision tree. Show the optimal tree as well as the tree resulting from the greedy top-down construction procedure. (Hint: my solution has three features, and the optimal tree uses two of them.)
2. Perceptron algorithm. Implement the Perceptron algorithm presented in class in 2 dimensions and perform the following experiments where concept C is defined by $C(\mathbf{x}) = +1$ if $x_1 + x_2 > 0$ and $C(\mathbf{x}) = -1$ otherwise.

Experiment 1:

Generate a 10 example training set by picking points uniformly at random from the unit circle and generating labels (y -values) according to C . Calculate the gap of the best homogeneous separating line (this gap is the distance between the separating line and the closest example, and the best separating line is not likely to be C 's decision boundary). Run the Perceptron algorithm and note how many "mistakes" it makes before finding a consistent hypothesis, and how many iterations through the data are required before it finds a consistent hypothesis.

Perform experiment 1 10 times and sort them by the gap. Do you see a relationship between the gap and the number of iterations or number of mistakes made?

Experiment 2:

Generate a 100 example training set by picking points uniformly at random from the unit circle, with noisy labels. For each example $\mathbf{x} = (x_1, x_2)$ in the training set, generate a random number r in $[0, 1]$. If $x_1 + x_2 + 2r - 1 > 0$ then set the label of \mathbf{x} to 1. If $x_1 + x_2 + 2r - 1 \leq 0$ then set the label of \mathbf{x} to -1 . Also generate a 100 example test set the same way.

This generates a noisy version of C where the noise tends to be concentrated around the decision boundary.

Run the following version of the perceptron algorithm for 500 iterations where each iteration uses a random point from the training set (rather than cycling through the training set) and save the weight vector \mathbf{w}_i after each iteration i . Compare how well the following prediction rules perform on the test set.

- (a) Last hypothesis: predict on \mathbf{x} with $\text{sign}(\mathbf{w}_{500} \cdot \mathbf{x})$, using the hypotheses from the last iteration.
- (b) Voted hypothesis: predict on \mathbf{x} using the majority of the \mathbf{w}_i , i.e. $\text{sign}\left(\sum_{i=1}^{500} \text{sign}(\mathbf{w}_i \cdot \mathbf{x})\right)$.
- (c) Longest survivor: Each \mathbf{w} values is created on some iteration t , predicts correctly for a while, and then makes a mistake at some later iteration t' . The survival time of \mathbf{w} is $t' - t$. Let \mathbf{w}_ℓ be the longest surviving hypothesis from the 500 iterations (pick the first one in case of ties, and assume that the last \mathbf{w} makes an incorrect prediction on iteration 501).

Create 10 different training and testing sets, and run the perceptron algorithm on each training set. How well do each of the three prediction rules do on the test sets?

Recommended exercises (not to be turned in):

- Make up a small dataset (say 8 examples with three boolean features) and compute the impurity (Equation 9.8) after each of the possible splits.
- Use Weka to learn a decision tree, neural network, and an SVM (Using the SMO algorithm) from the Iris2 dataset. It is also interesting to try training a simple XOR dataset (where $y = x_1 \oplus x_2$) with a neural network and seeing how slow the convergence is (especially when there are additional irrelevant features).
- Given one dimensional Gaussian distributions $P(x|+)$ and $P(x|-)$ having the same σ but different means, find the decision boundary minimizing the probability of a mistake when $P(+)=2P(-)$. Next, find the Bayes optimal decision boundary when the cost of a false positive is twice the cost of a false negative (and the cost of a correct prediction is 0).
- Dig out your old (linear?) algebra book and verify that equation (10.49) uses the Lagrange multipliers correctly. Also verify that the dual (10.52) is correctly formulated.