

8803 Machine Learning Theory

Homework # 4

Due: April 6th 2010

This homework is due by the start of class on April 6th. You can either submit the homework via the course page on T-Square or hand it in at the beginning of the class on April 6th.

Groundrules:

- Your work will be graded on correctness, clarity, and conciseness.
- You may collaborate with others on this problem set and consult external sources. However, you must *write your own solutions* and *list your collaborators/sources* for each problem.

Problems:

1. **SVM and 0-1 loss.** Suppose we have set $S = \{(x_1, \ell_1), \dots, (x_m, \ell_m)\}$ of labeled examples in R^n , and assume $|x_i| = 1$ for all i . It is NP-hard to find a linear separator that minimizes the number of points misclassified, so learning algorithms tend to optimize for other, related quantities. SVM in particular solves the optimization problem:

$$\begin{aligned} \text{minimize: } & |w|^2 + C \sum_i \epsilon_i \\ \text{subject to: } & \ell_i(w \cdot x_i) \geq 1 - \epsilon_i \text{ for all } i, \\ & \epsilon_i \geq 0 \text{ for all } i. \end{aligned}$$

Suppose that S has the property that the total distance one would need to move the points in order to make them separable by margin γ is d_γ . (By “total distance” we mean $\sum_i d_i$ where d_i is the distance that point x_i is moved. By “separable by margin γ ” we mean that for some hyperplane through the origin, all points are on the correct side and at distance at least γ from it.)

Show that for an appropriate value of C , the number of misclassifications made on S by the separator produced by SVM is at most $\frac{1}{2} + d_\gamma/\gamma$.

2. **Decision tree rank.** The *rank* of a decision tree is defined as follows. If the tree is a single leaf then the rank is 0. Otherwise, let r_L and r_R be the ranks of the left and right subtrees of the root, respectively. If $r_L = r_R$ then the rank of the tree is $r_L + 1$. Otherwise, the rank is the maximum of r_L and r_R .

Prove that a decision tree with ℓ leaves has rank at most $\log_2(\ell)$.

Extra Credit:

1. **Expressivity of decision lists** Show that the class of rank- k decision trees is a subclass of k -decision lists. (There are several different ways of proving this.)

Thus, we conclude that we can learn constant rank decision trees in polynomial time and arbitrary decision trees of size s in time and number of examples $n^{O(\log s)}$. (So this is almost a PAC-learning algorithm for decision trees.)