

# MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.867 MACHINE LEARNING, FALL 2009

## Problem Set 4

Due Date: Monday, November 9 (at 5pm)

NOTE: you will not receive a graded problem set in return for submitting one. We have adopted a new grading scheme for problem sets. We will help you understand and work through the problems. You are therefore strongly encouraged to clear up any misunderstandings prior to the deadline. By submitting your solutions, you indicate a) that you do understand the problem and b) that you have worked through the solution. Your submission will be recorded and you will receive no other grade from the submission. By completing a problem set, you demonstrate an achievement. In total, problem sets count 30% of the course grade.

Electronic submission is required! A submission form will be made available at the course website. You should have received the password required for submission via email.

### Problem 1 VC-dimension, margin

In this problem, we will investigate the VC-dimension of sets of classifiers. A *classifier* here is a function from some input space to the binary class labels  $+1, -1$ . We say that a set  $\mathcal{H}$  of classifiers *shatters* a set of points  $X = \{\underline{x}_1, \underline{x}_2, \dots, \underline{x}_n\}$  if we can classify the points in  $X$  in all possible ways. More precisely, for all  $2^n$  possible labeling vectors  $(y_1, y_2, \dots, y_n) \in \{-1, 1\}^n$ , there exists a classifier  $h \in \mathcal{H}$  such that  $h(\underline{x}_i) = y_i$  for all  $i$  (for any possible labeling of the points, there has to be a classifier in our set that reproduces those labels). It is important to understand that shattering is a property of a *set* of classifiers, not of a single classifier. A single classifier cannot shatter even a single point. The *VC-dimension* of a set  $\mathcal{H}$  of classifiers is the size of the largest set of points  $X$  that can be shattered by  $\mathcal{H}$ .

We often derive classifiers  $h(\underline{x})$  from real valued discriminant functions  $f(\underline{x})$  by taking the sign:  $h(\underline{x}) = \text{sign}(f(\underline{x}))$ . Although combinatorial properties such as shattering and VC-dimension are properties of sets of classifiers  $\mathcal{H}$ , through this mapping they can also be understood as properties of sets of discriminant functions  $\mathcal{F}$ .

**1.1** Decision stumps are not very complex classifiers. In fact, the VC-dimension of stumps in 1-dimensions is 2. They do not get much more complicated when we add dimensions. Show that the decision stumps can generate only  $2nd$  distinct labelings of  $n$  points in  $\mathcal{R}^d$  (Hint: look at how many ways stumps can label points in each dimension). Can you use this result to come up with an upper bound on the VC-dimension of stumps in  $\mathcal{R}^d$ ?

**1.2** We have already seen that combining stumps can generate much more complex classifiers. Indeed, ensembles with  $2n$  stumps can shatter any  $n$  distinct points in 1-dimensions. The construction also holds in  $\mathcal{R}^d$  under slightly weaker condition that each point has at least one coordinate value distinct from the other points. Thus the VC-dimension of ensembles with  $m$  stumps in  $\mathcal{R}^d$  is at least  $m/2$ . This is a lower bound. How complex can they be? Determine the VC-dimension of ensembles with two stumps in  $\mathcal{R}^2$ .

**1.3** Ensembles work quite well in practice despite their high VC-dimension. How can this be? We know that the boosting procedure tends to construct ensembles with large large voting margins on the training set.

In other words, after potentially many boosting iterations, we may find an ensemble such that

$$y_i h_m(\underline{x}_i) = y_i \sum_{j=1}^m \hat{\alpha}_j h(\underline{x}_i; \hat{\theta}_j) \geq \gamma_v, \quad i = 1, \dots, n \quad (1)$$

for some  $\gamma_v \in (0, 1]$ . We assume that the votes have already been normalized so that  $\sum_{j=1}^m \hat{\alpha}_j = 1$ . The larger the voting margin  $\gamma_v$  is, the better we can approximate  $h_m(\underline{x})$  with an ensemble that only has a few stumps (therefore low VC-dimension). Thus by looking for solutions that achieve a large voting margin we effectively operate in the space of classifiers with low VC-dimension. Let's make this a bit more precise.

We will approximate the ensemble  $h_m(\underline{x})$  by simply averaging a set of  $k$  stumps sampled at random from the  $m$  possible stumps according to the discrete distribution  $\hat{\alpha}_1, \dots, \hat{\alpha}_m$ . In other words, with probability  $\hat{\alpha}_j$  we include stump  $h(\underline{x}; \hat{\theta}_j)$  in the average, possibly multiple times. Our sampled approximation is then

$$\tilde{h}_k(\underline{x}) = \frac{1}{k} \sum_{l=1}^k h(\underline{x}; \hat{\theta}_{j_l}) \quad (2)$$

where  $j_l$  is the  $l^{\text{th}}$  stump we selected. Show that with high probability  $1 - \delta$ , the approximation  $\tilde{h}_k(\underline{x})$  still classifies all the  $n$  training points in the same way as  $h_m(\underline{x})$  provided that  $k$  is at least  $2(\log(1/\delta) + \log(n))/\gamma_v^2$ . Note that the number of stumps we need does not depend on  $m$ , i.e., the approximation works fine even if  $m = \infty!$ .

Useful tools include the union bound and

$$s \in \{-1, 1\}, \quad \text{Prob}\left(E\{s\} - \sum_{l=1}^k s_l/k \geq \gamma\right) \leq \exp(-\gamma^2 k/2) \quad (3)$$

where  $s_l$  are independent  $\pm 1$  samples with expectation  $E\{s\}$ .

## Problem 2 Probabilistic classifiers

**2.1** Suppose we use a probabilistic classifier where the class-conditional distributions are Gaussian. We will further restrict the class conditional distributions so that they have the same covariance matrix. As a result, the joint distribution of inputs  $\underline{x}$  and labels  $y$  is given by

$$P(\underline{x}, y; \theta) = q(y) N(\underline{x}; \underline{\mu}_y, \Sigma) \quad (4)$$

where  $q(y)$  is the prior probability of class  $y$ ,  $\underline{\mu}_y$  is the class dependent mean, and  $\Sigma$  is the common covariance matrix. The decision rule for this classifier is given by

$$\hat{y} = \arg \max_{y \in \{-1, +1\}} P(\underline{x}, y; \theta)$$

Since there are only two classes, we can obtain this rule by taking the sign of the discriminant function

$$f(\underline{x}; \theta) = \log \frac{P(\underline{x}, y = +1; \theta)}{P(\underline{x}, y = -1; \theta)} \quad (5)$$

Show that  $f(\underline{x}; \theta)$  is a linear classifier, i.e., that it can be written in the form  $\underline{\theta} \cdot \underline{x} + \theta_0$  for some  $\underline{\theta}$  and  $\theta_0$ . Define  $\underline{\theta}$  and  $\theta_0$  as a function of  $q(y)$ ,  $\underline{\mu}_y$ , and  $\Sigma$ .

**2.2** A Naive Bayes model is a simple yet commonly used probabilistic classifier over discrete features. The key identifying property of this model is that the feature values are independent of each other given the class. If the label is binary  $y \in \{-1, 1\}$  and each feature takes  $r$  possible discrete values in  $\{1, \dots, r\}$ , then the joint distribution over labels and features is given by

$$P(\underline{x}, y; \theta) = q(y) \prod_{j=1}^d q_j(x_j|y) \quad (6)$$

where  $\theta$  consists of the prior class probabilities  $q(y)$  and the class conditional feature distributions  $q_j(x_j|y)$ . How many independent parameters are there in the Naive Bayes model?

**2.3** According to the Naive Bayes model, the feature values are independent given the class. What about the marginal distribution  $P(\underline{x}; \theta)$ ? In other words, suppose we didn't observe the class label, are the features still independent? Provide an example using  $d = 2$  and  $r = 2$  that illustrates that the features  $x_1$  and  $x_2$  are indeed *not* independent in the marginal distribution

$$P(\underline{x}; \theta) = \sum_{y=-1,1} P(\underline{x}, y; \theta) = \sum_{y=-1,1} q(y)q_1(x_1|y)q_2(x_2|y) \quad (7)$$

Mixture distributions can be quite expressive even if constructed from simple components.

**2.4** We will consider here feature selection in a naive Bayes model. In particular, we will illustrate how the mutual information criterion discussed in earlier lectures is in a certain sense the optimal feature selection method for Naive Bayes. For this purpose, we define a restricted Naive Bayes model where only a *subset* of the features depend on the class. Let  $A$  denote the set of *active features* in the model. The set  $A$  is a subset of  $\{1, 2, \dots, d\}$ . The restricted model then takes the following form:

$$P(\underline{x}, y; \theta, A) = q(y) \prod_{j \in A} q_j(x_j|y) \prod_{j \notin A} q_j(x_j) \quad (8)$$

Note that for each active feature we generate the feature values in a class conditional fashion using  $q_j(x_j|y)$  while for inactive features we only specify  $q_j(x_j)$ .

Now, given labeled training data  $(\underline{x}_i, y_i)$ ,  $i = 1, \dots, n$ , we would like to find the subset of features to include so as to maximize the log-likelihood

$$l(D; \theta, A) = \sum_{i=1}^n \log P(\underline{x}_i, y_i; \theta, A) = \sum_{i=1}^n \log q(y_i) + \sum_{i=1}^n \sum_{j \in A} \log q_j(x_{ij}|y_i) + \sum_{i=1}^n \sum_{j \notin A} \log q_j(x_{ij})$$

with respect to  $\theta$  and  $|A| \leq k$ .

- a) The maximum likelihood estimates of parameters  $q(y)$ ,  $q_j(x_j|y)$  for  $j \in A$  and  $q_j(x_j)$  for  $j \notin A$  are given by

$$\hat{q}(y) = \frac{1}{n} \sum_{i=1}^n \delta(y_i, y), \quad \hat{q}_j(x_j|y) = \frac{1}{n\hat{q}(y)} \sum_{i=1}^n \delta(y_i, y) \delta(x_{ij}, x_j), \quad \hat{q}_j(x_j) = \frac{1}{n} \sum_{i=1}^n \delta(x_{ij}, x_j) \quad (9)$$

where  $\delta(z, z') = 1$  if  $z = z'$  and zero otherwise. Derive the answer for one of these.

- b) Fix  $A$  and consider moving any  $j \notin A$  into the active set. Show that the gain or improvement in the log-likelihood of the data that we would obtain as a result of making feature  $j$  active, i.e.,

$$\text{Gain}(j) = l(D; \hat{\theta}, A \cup \{j\}) - l(D; \hat{\theta}, A) \quad (10)$$

does not depend on  $A$ . In other words, if we move feature  $j$  from the inactive set to the active set, the improvement in the log-likelihood is the same regardless of which other features we have already made active. By  $l(D; \hat{\theta}, A)$  we mean the value of the log-likelihood evaluated using the maximum likelihood parameter values

$$l(D; \hat{\theta}, A) = \sum_{i=1}^n \log \hat{q}(y_i) + \sum_{i=1}^n \sum_{j \in A} \log \hat{q}_j(x_{ij}|y_i) + \sum_{i=1}^n \sum_{j \notin A} \log \hat{q}_j(x_{ij}) \quad (11)$$

(Hint: note that the estimated parameter values are not affected by  $A$ ).

- c) Show that the gain can be expressed in terms  $n$ ,  $\hat{q}(y)$ ,  $\hat{q}_j(x_j|y)$ , and  $\hat{q}_j(x_j)$  as

$$\text{Gain}(j) = n \sum_{y=\pm 1} \sum_{x_j=1}^r \hat{q}(y) \hat{q}_j(x_j|y) \log \frac{\hat{q}_j(x_j|y)}{\hat{q}_j(x_j)} \stackrel{\text{def}}{=} n I(X_j; Y) \quad (12)$$

where  $I(X_j; Y)$  is the *mutual information* between  $x_j$  and  $y$  measuring how much we learn about  $y$  by knowing the value of  $x_j$ . So, to find the best feature subset  $A$ , we can simply add features in the order of their gain. This information criterion does not specify how to select  $k$ . What would be a reasonable stopping criterion?