CS446: Pattern Recognition and Machine Learning **Fall 2009**

Final

December 11, 2009

This is a closed book exam. Everything you need in order to solve the problems is supplied in the body of the exam.

The exam ends at 11:00 am. It contains 5 problems.

You have 180 minutes to earn a total of 100 points. Answer each question in the space provided.

If you need more room, write on the back side of the paper and indicate that you have done so.

Clarity of writing is important — not just having the right answer. For full credit, you must show your work and explain your answers.

Good luck!

Name:

Problem 1 (20 points):

Problem 2 (20 points):

Problem 3 (20 points):

Problem 4 (20 points):

Problem 5 (20 points):

Total (100):

Problem 1 [PAC learning – 20 points]:

Consider the following learning task. Examples are points in the 3D space $(x, y, z) \in R^3$. The concept space C is the region between two spheres with center in the origin $(0, 0, 0)$. That is, each concept $f \in C$ is defined by two radius $0 < r_1 \leq r_2$. An example (x, y, z) is positive if and only if

$$r_1^2 \leq x^2 + y^2 + z^2 \leq r_2^2$$

In the learning scenario, we are given a collection of m examples labeled according to a hidden concept $f \in C$ and are required to learn a concept $h \in C$ that approximates f.

(a) Give an algorithm for learning this concept class. Explain it and prove its correctness. Write down explicitly the final hypothesis of your algorithm.

ANS:

$$U = \max_{i \text{ is positive}} x_i^2 + y_i^2 + z_i^2$$

$$L = \min_{i \text{ is positive}} x_i^2 + y_i^2 + z_i^2$$

The hypothesis is

$$L \leq x^2 + y^2 + z^2 \leq U$$

Proof: if there is a consistent hypothesis for this dataset, then distance between any positive examples need to be greater than L and smaller than U. No negative example should have this property. Therefore, the hypothesis we generated is correct.

(b) If there are several possible consistent hypotheses, choose one of them and argue using 1-2 sentences that it is likely to perform better than other possible hypotheses.

ANS:

$$U_p = \max_{i \text{ is positive}} x_i^2 + y_i^2 + z_i^2$$

$$L_p = \min_{i \text{ is positive}} x_i^2 + y_i^2 + z_i^2$$

Define $N_1 = \{i | i \text{ is negative}, x_i^2 + y_i^2 + z_i^2 \geq U_p\}$.

Define $N_2 = \{i | i \text{ is negative}, x_i^2 + y_i^2 + z_i^2 \leq L_p\}$.

$$U_n = \min_{i \in N_1} x_i^2 + y_i^2 + z_i^2$$

$$L_n = \max_{i \in N_2} x_i^2 + y_i^2 + z_i^2$$

The final hypothesis is

$$\frac{L_p + L_n}{2} \leq x^2 + y^2 + z^2 \leq \frac{U_p + U_n}{2}$$

We claim this is better because the "margin" is larger.

(c) Show that the concept class C is efficiently PAC learnable.

ANS:

First, the time complexity of the proposed algorithm is linear to the number of the examples.

Now we would like to show that the number example needed is also polynomial. We will prove this by showing the VC dimension is a constant. Moreover, the training examples needed for learning a concept is bounded by

$M = O(min\{\frac{1}{\epsilon}(\ln|H| + \ln 1/\delta), \frac{1}{\epsilon}(VC(H) + \ln 1/\delta)\}$

If $VC(C)$ is a constant, M is polynomial to the size of the problem. Therefore, we can claim that this is PAC learnable.

The VC dimension of the concept class C is 2. Here is the proof: 1) it is trivial that any two points which have different distance to the origin can be shattered by the class

2) For three points, a, b, c, assume that there distance to origin is r_a, r_b, r_c, respectively. Without loss the generality, assume $r_a \leq r_b \leq r_c$. If we assign a and c as positive examples, and assign b as a negative example. There is not consistent hypothesis for this dataset.

Therefore, the VC dimension is 2, so the concept class is PAC learnable.

(d) Assume now that there is some noise in the data, and the m examples may not be completely consistent with the target hypothesis. For simplicity, assume that no two examples have the same distance from the origin. Propose a learning algorithm that minimizes the number of mistakes made on the training data.

1) For the i-th point, collect $r_i = x_i^2 + y_i^2 + z_i^2$.

2) For $i = 1, \ldots, m$, $j = 1, \ldots, m$ try the following hypothesis

$$r_i \leq x^2 + y^2 + z^2 \leq r_j$$

And select the best one.

Problem 2 [Expectation-Maximization Algorithm - 20 points]

Consider the following generative probabilistic model:

$$W \rightarrow X \leftarrow Z$$

over the Boolean variables X, W, Z, where the data is generated according to:

- The variable W is set to 1 with probability α, and 0 with probability $1 - \alpha$.
- The variable Z is set to 1 with probability β, and 0 with probability $1 - \beta$.
- If $(W, Z) = (1, 1)$ then $X = 1$ with probability λ_{11}
  If $(W, Z) = (0, 1)$ then $X = 1$ with probability λ_{01}
  If $(W, Z) = (1, 0)$ then $X = 1$ with probability λ_{10}
  If $(W, Z) = (0, 0)$ then $X = 1$ with probability λ_{00}

This information is summarized below.

$$\begin{array}{rcl} P(W=1) & = & \alpha \\ P(Z=1) & = & \beta \\ P(X=1|W=1,Z=1) & = & \lambda_{11} \\ P(X=1|W=0,Z=1) & = & \lambda_{01} \\ P(X=1|W=1,Z=0) & = & \lambda_{10} \\ P(X=1|W=0,Z=0) & = & \lambda_{00} \end{array}$$

You need to estimate the parameters of this model. However, **one of the variables, Z, is not observed**. You are given a sample of m data points:

$$\{(w^{(j)}, x^{(j)})|w, x \in \{0, 1\}\}_{j=1}^{m}.$$

In order to estimate the parameters of the model, $\alpha, \beta, \boldsymbol{\lambda_{11}}, \boldsymbol{\lambda_{01}}, \boldsymbol{\lambda_{10}}, \boldsymbol{\lambda_{11}}$ you will derive update rules for them via the EM algorithm for the given model.

1. Express $P(w^{(j)}, x^{(j)})$ first in terms of conditional probabilities, and then in terms of the unknown parameters.

   ANS:

$$
\begin{aligned}
&P(w^{(j)}, x^{(j)}) = P(w^{(j)}, x^{(j)}, z^{(j)} = 0) + P(w^{(j)}, x^{(j)}, z^{(j)} = 1) \\
=&p(z^{(j)} = 0)P(w^{(j)})p(x^{(j)}|w^{(j)}), z^{(j)} = 0) + p(z^{(j)} = 1)P(w^{(j)})p(x^{(j)}|w^{(j)}), z^{(j)} = 1) \\
=&(1-\beta)[\alpha\lambda_{10}^{x_j}(1-\lambda_{10})^{1-x_j}]^{w_j}[(1-\alpha)\lambda_{00}^{x_j}(1-\lambda_{00})^{1-x_j}]^{1-w_j} + \\
&(\beta)[\alpha\lambda_{11}^{x_j}(1-\lambda_{11})^{1-x_j}]^{w_j}[(1-\alpha)\lambda_{01}^{x_j}(1-\lambda_{01})^{1-x_j}]^{1-w_j}
\end{aligned}
$$

2. Let $f_z^j = P(Z = z|w^{(j)}, x^{(j)})$, the probability that the hidden variable Z has value z.

   Express $f_1^{(j)}$ first in terms of conditional probabilities, and then in terms of the unknown parameters.

   ANS:

$$f_1^{(j)} = \frac{p(z^{(j)} = 1)p(w^{(j)})p(x^{(j)}|w^{(j)}), z^{(j)} = 1)}{p(z^{(j)} = 0)p(w^{(j)})p(x^{(j)}|w^{(j)}), z^{(j)} = 0) + p(z^{(j)} = 1)p(w^{(j)})p(x^{(j)}|w^{(j)}), z^{(j)} = 1)}$$
$$= \frac{(\beta)[\alpha\lambda_{11}^{x_j}(1-\lambda_{11})^{1-x_j}]^{w_j}[(1-\alpha)\lambda_{01}^{x_j}(1-\lambda_{01})^{1-x_j}]^{1-w_j}}{P(w^{(j)}, x^{(j)})}$$

3. Derive an expression for the expected log likelihood (LL), of the entire dataset, $\{(w^{(1)}, x^{(1)}), (w^{(2)}, x^{(2)}), \ldots, (w^{(m)}, x^{(m)})\}$ given the new parameter estimates $\tilde{\alpha}, \tilde{\beta}, \tilde{\lambda_{11}}, \tilde{\lambda_{10}}, \tilde{\lambda_{01}}, \tilde{\lambda_{00}}$. You can write your answer in terms of $f_1^{(j)}$ (even if you don't solve (b)).

   ANS:

$$\sum_{j=1}^{m} \left(f_1^j \log p(w^{(j)}, x^{(j)}, z^{(j)} = 1) + (1 - f_1^j) \log p(w^{(j)}, x^{(j)}, z^{(j)} = 0) \right)$$
$$= \sum_{j=1}^{m} f_1^j \log \left((\beta)[\alpha\lambda_{11}^{x_j}(1-\lambda_{11})^{1-x_j}]^{w_j}[(1-\alpha)\lambda_{01}^{x_j}(1-\lambda_{01})^{1-x_j}]^{1-w_j} \right)$$
$$+ \sum_{j=1}^{m} (1 - f_1^j) \log \left((1-\beta)[\alpha\lambda_{10}^{x_j}(1-\lambda_{10})^{1-x_j}]^{w_j}[(1-\alpha)\lambda_{00}^{x_j}(1-\lambda_{00})^{1-x_j}]^{1-w_j} \right)$$

4. Maximize the LL and determine the update rules for β and for λ_{10} according to the EM algorithm. **ANS:**

$$\frac{\partial LL}{\partial \beta} = 0 \Rightarrow \beta = \frac{\sum_{j=1}^{m} f_1^j}{m}$$

$$\frac{\partial LL}{\partial \lambda_{10}} = 0 \Rightarrow \lambda_{10} = \frac{\sum_{j=1}^{m}(1 - f_1^j) w_j x_j}{\sum_{j=1}^{m}(1 - f_1^j) w_j x_j + \sum_{j=1}^{m}(1 - f_1^j) w_j (1 - x_j)}$$

Problem 3 [SVM and Multiclass classification - 20 points]

Consider the following support vector machine formulation:

$$\min \quad \frac{1}{2} w^t w \tag{1}$$

$$s.t. \quad y_i(w^T x_i + b) \geq 1, \forall i = 1, \ldots, m, \tag{2}$$

where x_i represents the i-th example and $y_i \in \{-1, 1\}$ represents the label of x_i.

Assume that we have two learning algorithms (Algorithm 1 and Algorithm 2) that train a binary SVM with different time complexity: $O(dl^2)$ and $O(d^2 l)$, respectively, where l is the number of examples and d is number of features.

(a) The above SVM formulation assumes that the training data is linearly separable. Write down the SVM formulation which allows non-separability of the training data. Explain why the new formulation works.

$$\min \quad \frac{1}{2} w^t w + C \sum_{i=1}^{l} \xi_i \tag{3}$$

$$s.t. \quad y_i(w^T x_i + b) \geq 1 - \xi_i, \forall i = 1, \ldots, m, \tag{4}$$

(b) This SVM formulation, unfortunately, can only handle binary classification problems. Assume that we want to apply the binary SVM classifiers in a multicalss classification problem, which has m different classes and a total of l training instances. We will use two paradigms mentioned in class: all-vs-all (also known as "all pairs") and one-vs-all. Note that these two paradigms will use Algorithm 1 or Algorithm 2 as a "black box" algorithm to train the binary SVM models.

(b.1) Both all-vs-all and one-vs-all are training algorithms that allow us to extend binary classification algorithms to multicalss classification problems. Describe the training phase of these two techniques.

In order to be precise, describe the following details:

1. Number of times the black box Algorithm is being used
2. Number of examples used in each call to the black box Algorithm
3. The type of examples being used every time.

ANS:

skiped

(b.2) Assume that the l examples are split uniformly among the m classes and that Algorithm 1 (with time complexity $O(dl^2)$) is being used as the black box training algorithm.
Write down the *overall* training time complexity of using the all-vs-all and the 1-vs-all paradigms (in terms of m, l and d). Which training paradigm is more efficient?

ANS:

all vs all: $\binom{m}{2} d(\frac{2l}{m})^2 = O(dl^2)$ one vs all: $md(l)^2 = O(mdl^2)$
all vs all is more efficient

(b.3) Assume that the l examples are split uniformly among the m classes and that Algorithm 2 (with time complexity $O(d^2 l)$) is being used as the black box training algorithm.
Write down the *overall* training time complexity of using all-vs-all and 1-vs-all paradigms (in terms of m, l and d). Which training paradigm is more efficient?

ANS:
all vs all: $\binom{m}{2} d^2(\frac{2l}{m}) = O(md^2 l)$ one vs all: $md^2(l) = O(md^2 l)$
They are roughly the same

(c) Describe at least two ways the model trained with the all-vs-all paradigm can be used to classify a new example. Write down the overall time complexity (for one example) of the evaluation time of your algorithm.

ANS:

counting: do all predictions then counting: time complexity $O(m^2 d)$

dag: compare two classes at the same time, if one loss, never consider it again: time complexity $O(md)$

Problem 4 [Short questions - 20 points]

(a) Winnow [5 points]

There exist a linear function $f : X \to \{0, 1\}$ such that a data set consistent with it (that is, linearly separable) cannot be learned using the Winnow learning algorithm. Give an example for a function f with this property. How can you fix this problem, still using Winnow?

ANS:
Winnow can only learn the concept where all values of this concept is positive. For example, $x_1 - x_2 \geq 0$, cannot be learned by winnow. To solve this problem, we can try balanced winnow algorithm

(b) Boosting [8 points]

Let h_t be the hypothesis learned by the weak learner during the t^{th} round of Adaboost. Let D_t be the distribution over which h_t was learned, and let $D_t(i)$ be the weight assigned to the i^{th} example in that distribution. Then,

$$\epsilon_t = \sum_{i=1}^{m} D_t(i) I[h_t(x_i) \neq y_i]$$

is the error of h_t with respect to D_t, where m is the number of training examples, y_i is the label of example x_i, and $I[\cdot]$ is the indicator function which equals 1 if its argument is true and 0 otherwise. Adaboost then uses this ϵ_t to create D_{t+1}.

Recall that Adaboost updates the distribution in the following way:

$$D_{t+1}(i) = \begin{cases} \frac{D_t}{Z_t} 2^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ \frac{D_t}{Z_t} 2^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}, \quad Z_t \text{ is a normalization factor and } \alpha_t = \frac{1}{2} \log_2 \frac{1-\epsilon}{\epsilon}.$$

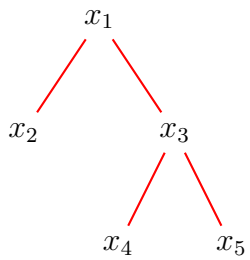
Compute Z_t as a function of ϵ_t. You need to show all of your work to get full credit of this problem.

ANS:

$$\begin{aligned} Z_t &= \sum_{i=1, h_t(x_i)=y_i}^{m} D_t(i) 2^{-\alpha_t} + \sum_{i=1, h_t(x_i)\neq y_i}^{m} D_t(i) 2^{\alpha_t} \\ &= \sum_{i=1, h_t(x_i)=y_i}^{m} D_t(i) (\frac{\epsilon}{1-\epsilon})^{\frac{1}{2}} + \sum_{i=1, h_t(x_i)\neq y_i}^{m} D_t(i) (\frac{1-\epsilon}{\epsilon})^{\frac{1}{2}} \\ &= (1-\epsilon)(\frac{\epsilon}{1-\epsilon})^{\frac{1}{2}} + (\epsilon)(\frac{1-\epsilon}{\epsilon})^{\frac{1}{2}} \\ &= 2(\epsilon)^{\frac{1}{2}}(1-\epsilon)^{\frac{1}{2}} \end{aligned}$$

(c) Learning Probability Distribution [8 points]

Consider a following generative model where the probability model is captured using the following tree:



Note that all variables here are binary variables. Therefore, $x_i \in \{0, 1\}$ for all $i = 1, \ldots, 5$.

(i) In order to finish the definition of this generative model, list conditional probabilities needed to be defined.

ANS:

$P(x_1), P(x_2|x_1), P(x_3|x_1), P(x_4|x_3), P(x_5|x_3)$

(ii) Write down an expression for $P(x_1, x_2, x_3|x_4 = 1, x_5 = 1)$ in terms of conditional probabilities defined in (i).

$$P(x_1, x_2, x_3|x_4 = 1, x_5 = 1) = \frac{P(x_1)P(x_2|x_1)P(x_3|x_1)P(x_4|x_3)P(x_5|x_3)}{\sum_{x_1,x_2,x_3} P(x_1)P(x_2|x_1)P(x_3|x_1)P(x_4|x_3)P(x_5|x_3)}$$

(iii) Assume that all the conditional probabilities you defined in (i) are equal to $\frac{1}{2}$, except that $P(x_4 = 1|x_3 = 1) = 1$ and $P(x_4 = 1|x_3 = 0) = 0$. What is the distance between the probability distributions $P(x_1, x_2, x_3|x_4 = 1, x_5 = 1)$ and the uniform distribution of $P(x_1, x_2, x_3)$ ($P(x_1, x_2, x_3) = \frac{1}{8}, \forall x_1, x_2, x_3$)?
(There is more than one way to define distance between distributions. You can choose any reasonable measure. Describe your choice clearly.)

ANS:

$P(x_1, x_2, x_3 = 0|x_4 = 1, x_5 = 1) = 0$

$P(x_1, x_2, x_3 = 1|x_4 = 1, x_5 = 1) = \frac{1}{4}$

the KL divergence is

$$\frac{1}{4}\log 2 + \frac{1}{4}\log 2 + \frac{1}{4}\log 2 + \frac{1}{4}\log 2 = 1$$

Problem 5 [Multinominal Distribution - 20 points]

In this problem, we will train a probabilistic model to classify a document written in a simplified language. However, instead of training the naive Bayes model with multivariate Bernoulli distributions, we will train it with a model that uses multinominal distributions.

Assume that all the documents are written in a language which has only three words a, b and c. All the documents have exactly n words (each word can be either a, b or c). We are given a labeled document collection $\{D_1, D, \ldots, D_m\}$. The label y_i of document D_i is 1 or 0, indicating whether D_i is "good" or "bad".

This model uses the multinominal distributions in the following way. Given the i-th document D_i, we denote by a_i (b_i, c_i, respectively) the number of times that word a (b, c, respectively) appears in D_i. Therefore, $a_i + b_i + c_i = |D_i| = n$.

In this model, we define

$$P(D_i|y=1) = \frac{n!}{a_i!b_i!c_i!}\alpha_1^{a_i}\beta_1^{b_i}\gamma_1^{c_i},$$

where α_1 (respectively β_1, γ_1) is the probability that word a (respectively b ,c) appears in a "good" document. Similarly,

$$P(D_i|y=0) = \frac{n!}{a_i!b_i!c_i!}\alpha_0^{a_i}\beta_0^{b_i}\gamma_0^{c_i},$$

where α_0 (respectively β_0, γ_0) is the probability that word a (respectively b ,c) appears in a "bad" document. Therefore, $\alpha_0 + \beta_0 + \gamma_0 = 1$.

(a) Given a document D, we want to classify it using $P(y|D)$. Write down an expression for $P(y|D)$ and explain what parameters are to be estimated from data in order to calculate $P(y|D)$.

ANS:

$P(y|D) = \frac{P(D|y)P(y)}{P(DP)} \propto P(D|y)P(y)$

Define $P(y) = \eta$, α_0,α_1,β_0,β_1,γ_0,γ_1

(b) **Derive** the most likely value of the parameters defined in (a).

The log likelihood

$$\sum \log P(y, D_i) = \sum_i y_i \eta \frac{n!}{a_i! b_i! c_i!} \alpha_1^{a_i} \beta_1^{b_i} \gamma_1^{c_i} + \sum_i (1 - y_i)(1 - \eta) \frac{n!}{a_i! b_i! c_i!} \alpha_0^{a_i} \beta_0^{b_i} \gamma_0^{c_i}$$

the answer

$$\eta = \frac{\sum_i y_i}{m}$$

$$\alpha_1 = \frac{\sum_i y_i a_i}{n \sum_i y_i}$$

$$\beta_1 = \frac{\sum_i y_i b_i}{n \sum_i y_i}$$

$$\gamma_1 = \frac{\sum_i y_i c_i}{n \sum_i y_i}$$

$$\alpha_0 = \frac{\sum_i (1 - y_i) a_i}{n \sum_i y_i}$$

$$\beta_0 = \frac{\sum_i (1 - y_i) b_i}{n \sum_i y_i}$$

$$\gamma_0 = \frac{\sum_i (1 - y_i) c_i}{n \sum_i y_i}$$

Some formula you may need:

1. $P(A, B) = P(A|B)P(B)$
2. $M = O(min\{\frac{1}{\epsilon}(\ln |H| + \ln 1/\delta), \frac{1}{\epsilon}(VC(H) + \ln 1/\delta)\}$