

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN  
Department of Electrical and Computer Engineering

ECE 598NA PATTERN RECOGNITION  
Fall 2006

**Final Exam**

Friday, December 15, 2006

- This is an **OPEN BOOK** exam. You may consult any non-human assistant for help, including your books, notes, calculator, or personal computer.
- You must **SHOW YOUR WORK** to get full credit.

Problem	Score
1	
2	
3	
4	
5	
6	
7	
Total	

**Name:** \_\_\_\_\_

**Problem 1 (10 points)**

$\hat{p}(x; \mathcal{D})$  is the Parzen window estimate of  $p(x)$ , the PDF of scalar random variable  $x$ , based on training dataset  $\mathcal{D}$ .  $\hat{p}(x; \mathcal{D})$  is computed as

$$\hat{p}(x; \mathcal{D}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{dx} K\left(\frac{x - x_i}{dx}\right)$$

where  $K(z)$  is the rectangular kernel, defined as follows in terms of the unit step function  $u(z)$ :

$$K(z) = u\left(z + \frac{1}{2}\right) - u\left(z - \frac{1}{2}\right)$$

Prove that

$$\lim_{dx \rightarrow 0} E(\hat{p}(x; \mathcal{D})) = p(x)$$

where the expectation is taken over  $\mathcal{D}$ , i.e.,

$$E(\hat{p}(x; \mathcal{D})) = \int \dots \int \hat{p}(x; \mathcal{D}) dx_1 \dots dx_n$$

Hint: notice that  $E(u(z)) = P(z \leq 0)$ .

**Problem 2 (20 points)**

$\hat{p}(x|\theta)$  is a mixture Gaussian probability density estimate,

$$\hat{p}(x|\theta) = \sum_{j=1}^c P(\omega_j) |\Sigma_j|^{-1/2} K\left(\Sigma_j^{-1/2}(x - \mu_j)\right)$$

where  $K(z) = (2\pi)^{-d/2} e^{-|z|^2/2}$  is the identity-covariance zero-mean Gaussian PDF, and  $\Sigma_j^{-1/2}$  is the whitening transform for pseudoclass  $\omega_j$ . The log-likelihood of a training database is defined to be

$$\mathcal{L}(\theta) = \sum_{i=1}^n \ln \hat{p}(x_i|\theta)$$

- (a) Suppose, for part (a) of this problem only, that the covariances and pseudoclass priors are identical and untrainable, i.e.,  $\Sigma_1 = \Sigma_2 = \dots = \Sigma_c = \text{constant}$ ,  $P(\omega_1) = \dots = P(\omega_c) = \frac{1}{c}$ . Prove that a local maximum of  $\mathcal{L}(\theta)$  is given by  $\mu_1 = \mu_2 = \dots = \mu_c = m$ , where  $m$  is the global data centroid.

- (b) Now suppose that the covariance matrices  $\Sigma_j$  are all different, and that they may be trained to maximize the likelihood. Specify the globally optimum value of  $\mathcal{L}(\theta)$ , and specify any particular parameter vector that achieves the global optimum.
- (c) Would you be happy to include your model from part (b) in a commercial product? Why or why not? (Assume an idealized competitive market, with informed consumers, in which only the most effective products are profitable).
- (d) Specify a mixture Gaussian training regimen sufficient to guarantee that, for any continuous true PDF  $p(x)$ , as  $n \rightarrow \infty$ ,  $\hat{p}(x|\theta) \rightarrow p(x)$ . Your training regimen should specify an algorithm (not necessarily a maximum likelihood algorithm!) for choosing  $c$  and  $[P(\omega_1), \dots, P(\omega_c), \mu_1, \dots, \mu_c, \Sigma_1, \dots, \Sigma_c]$  for any given value of  $n$ . Hint: turn the mixture

NAME: \_\_\_\_\_

*Final Exam*

Page 5

Gaussian estimate into a non-parametric PDF estimate.

**Problem 3 (10 points)**

The proximal distance between two sets is

$$D_{min}(\mathcal{S}_1, \mathcal{S}_2) = \min_{x_1 \in \mathcal{S}_1} \min_{x_2 \in \mathcal{S}_2} \|x_1 - x_2\|$$

Is  $D_{min}$  a metric? Prove your answer.

**Problem 4 (10 points)**

The diameter distance between two sets is

$$D_{max}(\mathcal{S}_1, \mathcal{S}_2) = \max_{x_1 \in \mathcal{S}_1} \max_{x_2 \in \mathcal{S}_2} \|x_1 - x_2\|$$

Is  $D_{max}$  a metric? Prove your answer.

**Problem 5 (15 points)**

Consider a data set containing scalar feature values  $x$ , each labeled with a scalar label  $y$ :

$$\mathcal{D} = \{(x_i, y_i)\} = \{(-1, 1), (0, -1), (1, 1)\}$$

Notice that there are two tokens from class  $y = 1$ , and one token from class  $y = -1$ . The feature values are  $x \in \{-1, 0, 1\}$ . You wish to train the bias coefficient of a one-node neural network in order to minimize the criterion

$$J = \sum_{i=1}^3 |f(x_i - b) - y_i|^2$$

using the following output nonlinearity:

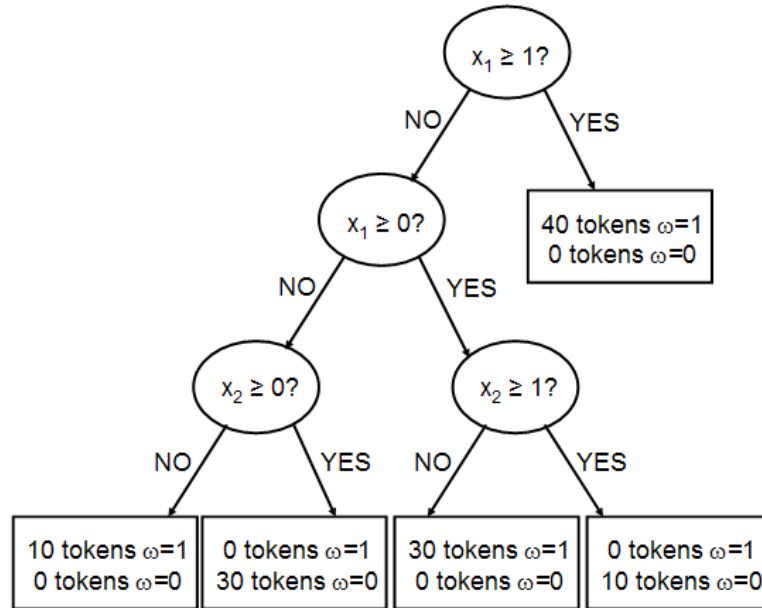
$$f(z) = \begin{cases} \text{sign}(z) & |z| \geq 1 \\ z & |z| \leq 1 \end{cases}$$

- (a) Find the global minimum value of  $J$ , and a bias coefficient  $b$  that achieves the global minimum.

- (b) What is the attractor basin for the bias coefficient that you specified in part (a)? That is, what is the set of initial coefficients  $b_0$  such that it is possible, via a sequence of infinitesimal moves in directions of decreasing  $J$ , to reach the bias coefficient that you specified in part (a)? In order to determine whether the region is closed or open, you may find it useful to define the derivative of the nonlinearity to be zero at its breakpoints, i.e.,  $f'(1) \equiv f'(-1) \equiv 0$ .

**Problem 6 (25 points)**

The following tree specifies a posterior probability function  $\hat{P}(\omega = 1|\vec{x})$ , as a function of the feature vector  $\vec{x} = [x_1, x_2]$ . Each node specifies the number of training tokens that fall into that node, and the class of those tokens, where the class is always either  $\omega = 0$  or  $\omega = 1$ . Assume frequency-based estimates of all branch and class probabilities, e.g.,  $\hat{P}(x_1 \geq 1) = 1/3$ , and  $\hat{P}(\omega = 1|x_1 \geq 1) = 1$ .



- (a) Suppose, for parts (a) and (b) of this problem, that the feature  $x_2$  is unavailable, and therefore the tree must be trimmed back to use only the top two questions. The resulting tree estimates a posterior probability function  $\hat{P}(\omega = 1|x_1)$ . Draw  $\hat{P}(\omega = 1|x_1)$  as a function of  $x_1$ .

(b) The entropy of  $\omega$  given  $x_1$  is defined to be

$$H(\omega|x_1) = - \int_{-\infty}^{\infty} p(x_1) \sum_{j=0}^1 P(\omega = j|x_1) \log_2 P(\omega = j|x_1) dx_1$$

Estimate  $H(\omega|x_1)$  using the same tree that you used in part (a). You may find it useful to approximate  $\log_2(3) \approx 1.6$ .

(c) Parts (c)-(e) of this problem use the complete tree, including both  $x_1$  and  $x_2$  as inputs (four question nodes). The complete tree specifies a deterministic posterior PMF  $\hat{P}(\omega = 1|\vec{x}) \in \{1, 0\}$ . Draw the decision boundary, in  $\vec{x}$ , between the regions  $\hat{P}(\omega = 1|\vec{x}) = 1$  and

$$\hat{P}(\omega = 1|\vec{x}) = 0.$$

- (d) Design a neural network that computes the same posterior PMF as the classification tree in part (c). Your network should have three linear input nodes ( $x_1$ ,  $x_2$ , and bias). It should have one output node, whose value is  $\hat{P}(\omega = 1|\vec{x})$ . You may use up to two hidden layers. Assume a unit step nonlinearity for the hidden and output nodes,  $f(z) = u(z)$ .

Specify all connection weights and biases.

- (e) What would be the effect on the function  $\hat{P}(\omega = 1|\vec{x})$  calculated by your neural network if the nonlinearities in the hidden and output nodes were replaced by sigmoids,  $f(z) = (1 + e^{-z})^{-1}$ ? (Assume that none of the connection weights or biases are changed).

**Problem 7 (10 points)**

In this problem, you will devise an algorithm for minimum-classification-error (MCE) linear discriminant analysis. You are given the following training dataset:

$$\mathcal{D} = \{(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)\}$$

where  $\vec{x}_i$  is a feature vector drawn from some unknown and extremely complicated PDF, and  $y_i \in \{-1, 1\}$  is the true class label of token  $\vec{x}_i$ . Design a linear feature extraction algorithm

$$\xi_i = \vec{w}^T \begin{bmatrix} 1 \\ \vec{x}_i \end{bmatrix}$$

that computes the new feature  $\xi_i$  in order to minimize training-corpus overlap of the two classes. In other words, you want to find the following  $\vec{w}$ :

$$\vec{w} = \arg \min J, \quad J = \sum_{i=1}^n u(-y_i \xi_i)$$

where  $u(z)$  is the unit step function.

- (a) Devise an algorithm that will estimate  $\vec{w}$ . If your algorithm requires eigenvalue analysis of a matrix, specify the matrix and the ordering of the eigenvalues. If your algorithm requires gradient descent, specify the optimality criterion and its gradient.

- (b) Does your algorithm in part (a) compute a global minimum of  $J$ , or a local minimum? If it computes a global minimum, prove that this minimum is global. If it computes a local minimum, suggest a heuristic that will help you make sure that you don't end up with a bad local minimum.