Practice Midterm Exam

EECS 545: Machine Learning

Winter, 2016

Name:

UM uniqname:

- **Closed book. Three sheets of paper of notes are allowed. No computers, cell phones or calculators.**
  - Showing your work makes partial credit possible.
    If you write nothing at all, it's hard to justify any score but zero.
  - Feel free to use the backs of the sheets for scratch paper.
  - Write clearly. If we can't read your writing, it will be marked wrong.
- This course operates under the rules of the College of Engineering Honor Code. Your signature endorses the pledge below. **After** you finish your exam, please sign below:
  I have neither given nor received aid on this examination, nor have I concealed any violations of the Honor Code.

Problem 1 (True/False). Are the following statements true or false? (No need for explanations unless you feel the question is ambiguous and want to justify your answer).

1. The error on the training set is a better estimate of the generalization error than the error on the test set.

   False. The error on the training set is typically quite biased, since the hypothesis was fit to this data.

2. The perceptron algorithm finds the maximum margin classifier if the data is linearly separable.

   False. It only find *some* classifier which separates the data.

3. Bayesian reasoning is popular since it avoids the need to explicitly specify a prior distribution.

   False. Priors are a Bayesian concept!

4. Assume we have trained a model for linear discriminant analysis, and we obtained parameters Σ, the covariance matrix, and μ_1, μ_2, the class means. We learned in class that the decision boundary between classes $c = 0$ and $c = 1$, i.e. the set $\{\mathbf{x} : P(y = c|\mathbf{x}, \Sigma, \mu_0, \mu_1) = 0.5\}$, is linear in the input space. But it is not linear at thresholds other than 0.5; for example, the set $\{\mathbf{x} : P(y = c|\mathbf{x}, \Sigma, \mu_1, \mu_2) = 0.9\}$ is not an affine subspace.

   False. Assume $c = 0$. We can write the set $\{\mathbf{x} : P(y = 0|\mathbf{x}, \Sigma, \mu_1, \mu_2) = q\}$ for any q as the set of $\mathbf{x}$ satisfying

$$\frac{\exp(-\frac{1}{2}(\mu_0 - \mathbf{x})^\top \Sigma^{-1}(\mu_0 - \mathbf{x}))}{\exp(-\frac{1}{2}(\mu_0 - \mathbf{x})^\top \Sigma^{-1}(\mu_0 - \mathbf{x})) + \exp(-\frac{1}{2}(\mu_1 - \mathbf{x})^\top \Sigma^{-1}(\mu_1 - \mathbf{x}))} = q.$$

   If we simplify this further we get

$$\exp(-\frac{1}{2}(\mu_1 - \mathbf{x})^\top \Sigma^{-1}(\mu_1 - \mathbf{x})) + \frac{1}{2}(\mu_0 - \mathbf{x})^\top \Sigma^{-1}(\mu_0 - \mathbf{x}))) = \frac{1}{q} - 1.$$

   If you take the log of both sides, and you cancel the $\mathbf{x}^\top \Sigma^{-1} \mathbf{x}$ terms, you obtain a linear equation in terms of $\mathbf{x}$.

5. Locally-weighted linear regression can produce nonlinear fits to the data.

   True

6. The specification of a probabilistic discriminative model can often be interpreted as a method for creating new, "fake" data.

   False. This is true for generative probabilistic models only.

7. Gaussian Discriminant Analysis as an approach to classification cannot be **applied** if the true class-conditional density for each class is *not* Gaussian.

   False. We can apply maximum likelihood (or MAP) estimation methods regardless of whether the data were truly drawn from these particular distributions.

8. Linear Regression can only be applied when the target values are binary or discrete.

   False. Linear regression requires real-valued targets.

9. The soft-margin SVM tends to have larger margin when the parameter C increases.

   False. When the parameter C increases, the objective function puts greater weight on the misclassification costs (the ξ terms). This reduces pressure on minimizing $\|w\|^2$, which is equivalent to maximizing the margin. Thus the resulting SVM solution may return a w with an even smaller margin (indeed increasing C will certainly not *increase* the margin size).

10. To solve

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \quad s.t. \quad \begin{array}{l} \sum_{i=1}^n x_i = 1 \\ \mathbf{x}_i \geq 0,\ \forall i \end{array}$$

   The Lagrangian would be $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \nu) = f(\mathbf{x}) + \boldsymbol{\lambda}^T\mathbf{x} + \nu(\mathbf{1}^T\mathbf{x} - 1)$ where $\boldsymbol{\lambda} \in \mathbb{R}^n$, $\lambda_i \geq 0$, $\nu \in \mathbb{R}$, and $\mathbf{1}$ is a vector of length n of all 1s.

   False. $\lambda^T\mathbf{x}$ should be negative

Problem 2 (Probability). For data D and hypothesis H, say whether or not the following equations must always be true.

a $\sum_h P(H = h|D = d) = 1$

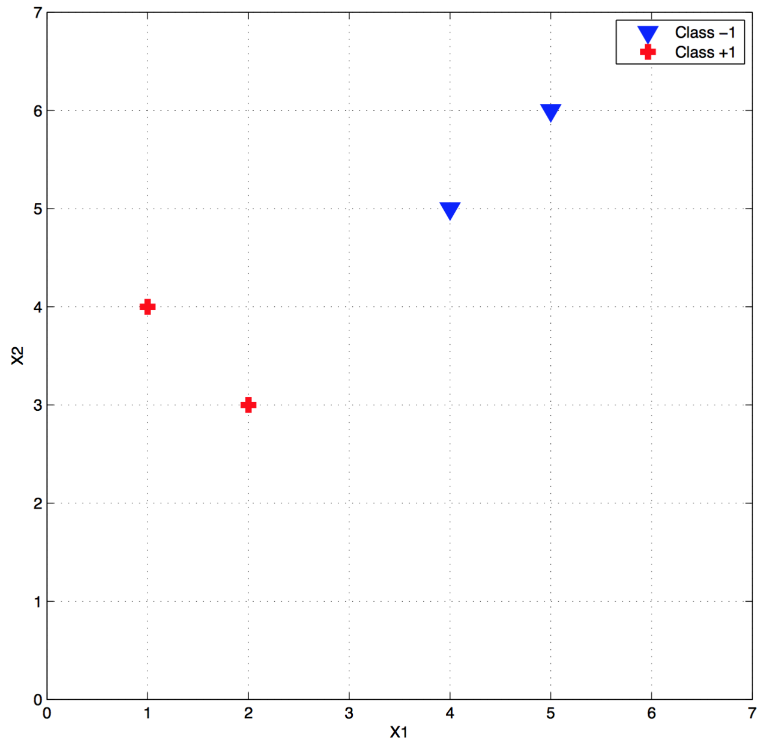
Yes

b $\sum_h P(D = d|H = h) = 1$

No

c $\sum_h P(D = d|H = h)P(H = h) = 1$

No



Problem 3 (SVM). 1. In class we learnt that SVM can be used to classify linearly inseparable data by transforming it to a higher dimensional space with a kernel $K(x, z) = \phi(x)^T \phi(z)$, where $\phi(x)$ is a feature mapping. Let K_1 and K_2 be $R^n \times R^n$ kernels, K_3 be a $R^d \times R^d$ kernel and $c \in R^+$ be a positive constant. $\phi_1 : R^n \to R^d$, $\phi_2 : R^n \to R^d$, and $\phi_3 : R^d \to R^d$ are feature mappings of K_1, K_2 and K_3 respectively. Explain how to use ϕ_1 and ϕ_2 to obtain the following kernels.

a $K(x, z) = cK_1(x, z)$

$\phi(x) = \sqrt{c}\phi_1(x)$

b $K(x, z) = K_1(x, z)K_2(x, z)$

$\phi(x) = \phi_1(x)\phi_2(x)$

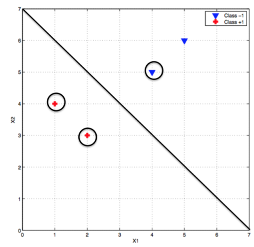
2. Support vector machines learn a decision boundary leading to the largest margin from both classes. You are training SVM (with no slack) on a tiny dataset with 4 points shown in Figure above. This dataset consists of two examples with class label -1 (denoted with plus), and two examples with class label $+1$ (denoted with triangles).

   (a) Find the weight vector w and bias b. What is the equation corresponding to the decision boundary?

SVM tries to maximize the margin between two classes. Therefore, the optimal decision boundary is diagonal and it crosses the point (3,4). It is perpendicular to the line between support vectors (4,5) and (2,3), hence it is slope is m = -1. Thus the line equation is $(x_2 - 4) = -1(x_1 - 3)$ and $x_1 + x_2 = 7$. From this equation, we can deduce that the weight vector has to be of the form (w_1, w_2), where $w_1 = w_2$. It also has to satisfy the following equations: $2w_1 + 3w_2 + b = 1$ and $4w_1 + 5w_2 + b = -1$ Hence $w_1 = w_2 = -0.5$ and $b = 3.5$

(b) Circle the support vectors and draw the decision boundary in Figure above.

Figure 1:



Problem 4 (Coin Flips and Pseudocounts). Suppose we flip a (not necessarily fair) coin N times and wish to estimate its bias θ after observing X heads. We endow θ with a Beta prior. Mathematically, our model is

$$\theta \sim \text{Beta}(a, b)$$
$$X \sim \text{Binomial}(N, \theta)$$

Part A. Derive the maximum likelihood estimate $\hat{\theta}_{ML}$ of the coin's bias? Show your work.

(1) The binomial likelihood function is

$$P(x|\theta) = \binom{N}{x} \theta^x (1-\theta)^{N-x}$$

(2) We seek the parameter maximizing this likelihood, or equivalently, maximizing its logarithm,

$$\hat{\theta}_{ML} = \arg\max_\theta P(X|\theta) = \arg\max_\theta \log P(X|\theta)$$

(3) The log-likelihood function is

$$\ell(\theta|x) = \log P(x|\theta) = \log \binom{N}{x} + x \log \theta + (N-x) \log(1-\theta)$$

(4) The derivative of this function is

$$\frac{\partial \ell}{\partial \theta} = \frac{x}{\theta} - \frac{N-x}{1-\theta}$$

(5) Setting this derivative to zero, we obtain the maximum likelihood estimate,

$$\hat{\theta}_{ML} = \frac{x}{N}$$

Part B. Write down the corresponding MAP estimate $\hat{\theta}_{MAP}$. No need to show your work.

$$\hat{\theta}_{MAP} = \frac{x+a-1}{N+a+b-2}$$

Problem 5 (Irrelevant Features with Naive Bayes). In this exercise, we consider words that are *nondiscriminative* for document classification (such as 'the', 'and', etc.) and analyze their impact on the decision made by Naive Bayes in several settings.

Let $x_{dw} = 1$ if word w occurs in document d and $x_{dw} = 0$ otherwise. Let the vocabulary size be W, and let θ_{cw} be the estimated probability $P(x_{dw} = 1|c)$ that word w occurs in documents of class c. Recall that the joint likelihood for Naive Bayes is

$$P(\mathbf{x}_d, c|\theta) = P(\mathbf{x}_d|c, \theta) = P(c) \prod_{w=1}^{W} P(x_{dw}|\theta_{cw})$$

where $P(c)$ specifies the class priors, and $\mathbf{x}_d = (\mathbf{x}_{d1}, \dots, \mathbf{x}_{dW})$ is a document.

<u>Part A.</u> Here, we show that Naive Bayes is a linear classifier. Define the new parameter vector

$$\beta_c = \left(\log \frac{\theta_{c1}}{1 - \theta_{c1}}, \cdots, \log \frac{\theta_{cW}}{1 - \theta_{cW}}, \sum_{w=1}^{W} \log(1 - \theta_{cw}) \right)^T$$

and let $\phi(\mathbf{x}_d) = (x_{d1}, \dots, x_{dW}, 1)^T$. Show that $\log P(\mathbf{x}_d|c, \theta) = \phi(\mathbf{x}_d)^T \beta_c$.

(1) The log-likelihood that document $\mathbf{x}$ belongs to class c is

$$\begin{aligned}
\log P(\mathbf{x}_d|c, \theta) &= \log \prod_{w=1}^{W} P(x_{dw}|c, \theta) \\
&= \log \prod_{w=1}^{W} \theta_{cw}^{x_{dw}} (1 - \theta_{cw})^{x_{dw}} \\
&= \sum_{w=1}^{W} x_{dw} \log \theta_{cw} + (1 - x_{dw}) \log(1 - \theta_{cw}) \\
&= \sum_{w=1}^{W} x_{dw} \log \frac{\theta_{cw}}{1 - \theta_{cw}} + \sum_{w=1}^{W} \log(1 - \theta_{cw})
\end{aligned}$$

(2) We can write this more succinctly as

$$\log P(\mathbf{x}_d|c, \theta) = \phi(\mathbf{x}_d)^T \beta_c$$

where $\mathbf{x}_d = (x_{d1}, \dots, x_{dW})$ is a bit vector, $\phi(\mathbf{x}_d) = (\mathbf{x}_d, 1)$, and

$$\beta_c = \left(\log \frac{\theta_{c1}}{1 - \theta_{c1}}, \cdots, \log \frac{\theta_{cW}}{1 - \theta_{cW}}, \sum_{w=1}^{W} \log(1 - \theta_{cw}) \right)^T$$

(3) Naive Bayes is a linear classifier because the class-conditional density is a linear function (inner product) of the parameters β_c.

Part B. Suppose there are only two possible document classes c_A and c_B, and assume a uniform class prior $\pi_A = \pi_B = 0.5$. and find an expression for the log posterior odds ratio R, shown below, in terms of the features $\phi(\mathbf{x}_d)$ and the parameters β_1 and β_2.

$$R = \log \frac{P(c_A|\mathbf{x}_d)}{P(c_B|\mathbf{x}_d)}$$

(1) The **posterior odds ratio** is the ratio of posterior class probabilities. If $R > 0$, class A is more likely than class B, and so we choose to classify document $\mathbf{x}_d$ as class A. If $R < 0$, we do the opposite.

(2) Given our uniform class prior, applying Bayes' Rule yields instead a ratio of likelihoods

$$R = \log \frac{P(\mathbf{x}_d|c_A)P(c_A)}{P(\mathbf{x}_d|c_B)P(c_B)} = \log \frac{P(\mathbf{x}_d|c_A)}{P(\mathbf{x}_d|c_B)}$$

(3) Using our result from the previous part, we arrive at the solution:

$$\boxed{R = \log P(\mathbf{x}_d|c_A) - \log P(\mathbf{x}_d|c_B) = \phi(\mathbf{x}_d)^T [\beta_1 - \beta_2]}$$

Part C. Intuitively, words that occur in both classes are not very *discriminative*, and therefore should not affect our beliefs about the class label. State the conditions under which the presence or absence of a particular word w in a test document will have no effect on the class posterior (such a word will effectively be ignored by the classifier).

(1) It makes sense that words with an equal probability of appearing in either class would be nondiscriminative. Let's prove this. Suppose for some word w that $\theta_{Aw} = \theta_{Bw}$.

(2) How does the inclusion of w in a document change the odds ratio R? Consider two documents $\mathbf{x}$ and $\mathbf{y}$, identical except that $x_w = 0$ and $y_w = 1$, i.e. word w appears in $\mathbf{y}$ but not $\mathbf{x}$. Then,

$$\begin{aligned} R_y - R_x &= \phi(\mathbf{x})^T [\beta_1 - \beta_2] - \phi(\mathbf{y})^T [\beta_1 - \beta_2] && (1) \\ &= [\phi(\mathbf{x}) - \phi(\mathbf{y})]^T [\beta_1 - \beta_2] && (2) \end{aligned}$$

(3) Recall that $\mathbf{x}$ and $\mathbf{y}$ are bit vectors, identical in all but one coordinate. So, the quantity $\phi(\mathbf{x})^T - \phi(\mathbf{y})^T$ is all zeros except in position w, and so, recalling that $\theta_{Aw} = \theta_{Bw}$, we have:

$$R_y - R_x = \log \frac{\theta_{Aw}}{1 - \theta_{Aw}} - \log \frac{\theta_{Bw}}{1 - \theta_{Bw}} = \log \frac{\theta_{Aw}}{\theta_{Bw}} \frac{1 - \theta_{Aw}}{1 - \theta_{Bw}} = 0 \qquad (3)$$

(4) The log posterior odds ratio for $\mathbf{x}$ and $\mathbf{y}$ are the same! This means that their classification will be the same, confirming our intuition.

(5) It is worth noting that $R_y - R_x$ is zero only if the condition $\theta_{Aw} = \theta_{Bw}$ holds. (*Prove it!*)

Part D. Consider a set of documents $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with labels $\mathcal{Y} = \{y_1, \ldots, y_n\}$. Suppose a particular word w always occurs in every document, regardless of class. Let there be N_A and N_B documents in classes A and B respectively, where $N_A \neq N_B$ (class imbalance). If we estimate the parameters θ_{cw} with the posterior mean under a uniform Beta$(1, 1)$ prior after observing data $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$, will word w be ignored by our classifier?

(1) The parameters θ for our Naive Bayes model are computed by simply partitioning the documents $\mathcal{X}$ by class and fitting a per-class Beta-Binomial model to find the per-class parameters θ_c.

$$\theta_{cw} = \frac{1 + \sum_{\mathbf{x}_k : y_k = c} x_{kw}}{2 + N_c} \tag{Murphy Eqn. 3.23}$$

(2) Now, the question becomes whether this estimate satisfies the conditions in the last part. Under the assumption of class imbalance, where $N_A \neq N_B$, we see clearly that it does not,

$$\theta_{Aw} = \frac{1 + N_A}{2 + N_A} \neq \frac{1 + N_B}{2 + N_B} = \theta_{Bw} \tag{4}$$

noting that x_{kw} is always equal to one, since word w appears in every document.

(3) So, word w is not ignored by our classifier if we place a prior on the parameters θ! The more drastic the class imbalance, the more the presence of w influences the classification.

(4) This phenomenon is easy to explain. Recall that in a Beta-Binomial model, the Beta hyperparameters can be interpreted as **pseudocounts** (see Murphy §3.3 for a review). Using a Beta(α, β) prior encodes the assumption that we have already seen α and β examples of word w in classes A and B respectively!

Problem 6 (Convexity). Let $J(\boldsymbol{\theta})$ be a twice-differentiable function such that

$$\nabla^2 J(\boldsymbol{\theta}) \preceq B$$

i.e., $B - \nabla^2 J(\boldsymbol{\theta})$ is positive semi-definite for some fixed positive definite matrix B (independent of $\boldsymbol{\theta}$). Show that given a fixed value $\boldsymbol{\theta}^{(t)}$, the function

$$J_t(\boldsymbol{\theta}) = J(\boldsymbol{\theta}^{(t)}) + \nabla J(\boldsymbol{\theta}^{(t)})^T(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)})^T B(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)})$$

is a majorizing function of $J(\boldsymbol{\theta})$; i.e., for all $\boldsymbol{\theta}$, $J_t(\boldsymbol{\theta}) \geq J(\boldsymbol{\theta})$, and $J_t(\boldsymbol{\theta}^{(t)}) = J(\boldsymbol{\theta}^{(t)})$.

Hint: A twice continuously differentiable function f admits the quadratic expansion

$$f(\mathbf{x}) = f(\mathbf{y}) + \langle \bigtriangledown f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{1}{2}\langle \mathbf{x} - \mathbf{y}, \bigtriangledown^2 f(\mathbf{y} + t(\mathbf{x} - \mathbf{y}))(\mathbf{x} - \mathbf{y}) \rangle$$

for some $t \in (0, 1)$.

$$\begin{aligned} J_t(\boldsymbol{\theta}) - J(\boldsymbol{\theta}) &= \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)})^T(B - \nabla^2 J(\boldsymbol{\theta}^{(t)} + t(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)})))(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}) \\ &\geq 0 \end{aligned}$$

Obviously, $J_t(\boldsymbol{\theta}^{(t)}) = J(\boldsymbol{\theta}^{(t)})$.

Problem 7 (Logistic Regression). Assume we have a training dataset that is linearly separable. Assume we train a logistic regression on this dataset with fixed parameters (we use the standard sigmoid function). Our logistic regression function predicts a probability for each new example, but assume we convert this to a classifier by thresholding the probability at $p \geq 0.5$ and $p < 0.5$. Question: if we measured this error on the training set, is it guaranteed that this error is zero?

Either prove that it does have zero training error or propose a dataset where the logistic regression returns a classifier which has non-zero training error.

When the training data are linearly separable, logistic regression does not even have a solution!

Consider a dataset that is linearly separable and consider some vector w that separates the data. For simplicity assume that for x_i in class 0 we have $w^\top x_i < 0$ and for any x_j in class 1 we have $w^\top x_j > 0$ (that is, assume no offset). The negative log likelihood function for logistic regression is

$$\sum_{i \in \text{ class } 0} \log(1 + \exp(w^\top x_i)) + \sum_{j \in \text{ class } 1} \log(1 + \exp(-w^\top x_j))$$

Notice that we can scale w by any number larger than 1 and *decrease* the objective function. Thus, the solution to Logistic Regression does not exist.

(PS This question was written by Prof Jake, and I intended initially to write it slightly differently. The correct answer is above, and yes it does look like a bit of a trick question. Whoops! We'll be nicer on the actual exam :-))