A

Midterm Exam
GU4241/GR5241 Spring 2017

Name _____

UNI _____

Problem 0: UNI (2 points)

Write your name and UNI on the first page of the problem sheet. After the exam, please return the problem sheet to us.

Problem 1: Short questions (2+2+3+3+4+4 points)

Short answers (about one sentence) are sufficient.

(a) (**Yes/No**) Is the Bayes classifier always the optimal one if we know the joint distribution of the predictors and the response? Explain briefly.

(b) A natural cubic spline with K knots is a function that:

- is a cubic polynomial between each pair of knots,
- is linear beyond the boundary knots,
- is continuous at the knots, and
- has continuous first and second derivatives at each knot.

Without using the definition of a natural cubic spline in terms of basis functions, explain why the splines has K free parameters or degrees of freedom.

(c) Consider the convex optimization problem over $x \in \mathbb{R}^2$:

$$\begin{array}{ll} \text{min} & f(x) \\ \text{s.t.} & g(x) \leq 0 \end{array}$$

Consider a point x^* with $\nabla f(x^*) = (0,0)^T$ and $g(x^*) \leq 0$. Is x^* a minimum? Explain briefly.

(d) Consider the same optimization problem, if we have point x^* with $\nabla f(x^*) = \begin{bmatrix} 0.9 \\ 1.2 \end{bmatrix}$ and $\nabla g(x^*) = \begin{bmatrix} -1.8 \\ -2.4 \end{bmatrix}$. Is x^* a minimum? Explain briefly.

(e) Assume that our model is $y = \mathbf{x}^T\beta_0 + \epsilon$, where $\mathbf{x}$ is a p-dimensional vector and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ with KNOWN σ^2. We fit ridge regression on a training set $(\mathbf{X}, \mathbf{y})$ with n observations:

$$\hat{\beta}_{\text{ridge}} = \underset{\beta}{\text{argmax}}\ (\mathbf{y} - \mathbf{X}\beta)^T(y - \mathbf{X}\beta) + \lambda\|\beta\|_2^2$$

For any fixed λ, write down the AIC and BIC statistics of the model.

(f) We want to perform clustering analysis on a data set in which about 5% of the data can be expected to be extreme outliers due to errors in the measuring device. Which one of the following two algorithms do you think will perform better on this data set, K-means or K-medoids? Explain briefly.

Solution:

(a) No. The Bayes classifier is only optimal under the 0-1 loss.

(b) If there were no constraints, we would have 4 parameters for each cubic piece, for a total of $4(K+1)$ parameters. However, we have 3 constraints at each knot and each one eliminates one degree of freedom. Beyond the boundaries, we have extra 2 constraints on for each side. This leaves us with $4(K+1) - 3K - 2 \times 2 = K$ parameters.

(c) Yes, this is the case when the minimum is either achieved at interior of the feasible set ($g(x^*) < 0$), or the global minimum can be obtained at the boundary ($g(x^*) = 0$).

(d) The answer depends. If $g(x^*) = 0$, then the minimum is achieved at the boundary; if not, then x^* is not the solution.

(e) The fitted value can be obtained by

$$\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}.$$

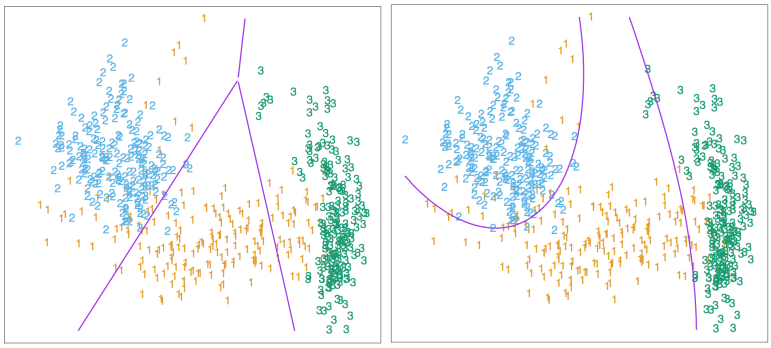
Denote $\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T$ by $\mathbf{S}_\lambda$, then

$$\begin{aligned} AIC(\lambda) &= \frac{1}{n}(\hat{\mathbf{y}} - \mathbf{y})^T(\hat{\mathbf{y}} - \mathbf{y}) + 2 \cdot \frac{\mathsf{tr}(\mathbf{S}_\lambda)}{n}\sigma^2 \\ BIC(\lambda) &= \frac{1}{n}(\hat{\mathbf{y}} - \mathbf{y})^T(\hat{\mathbf{y}} - \mathbf{y}) + (\log n) \cdot \frac{\mathsf{tr}(\mathbf{S}_\lambda)}{n}\sigma^2. \end{aligned}$$

(f) K-medoids will be better, since it requires the centroid to be an observation. By requiring this, K-medoids becomes more robust to outliers compared to K-means.

Problem 2: Decision boundaries (5+5 points)

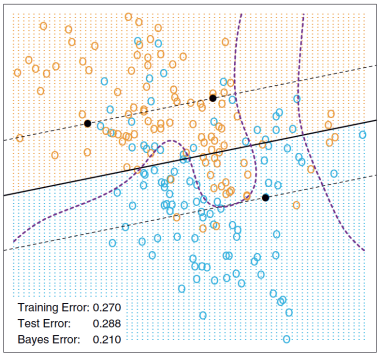
(a) We have some two-dimensional data from three classes as shown in the plots. The following two pictures, which one is more likely to be the output of QDA?



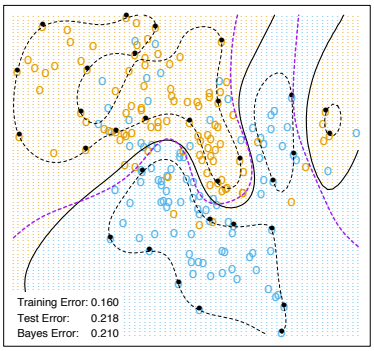
(I) (II)

Solution: (II) is more likely to be the output of QDA, since the decision boundary in figure (II) is non-linear.

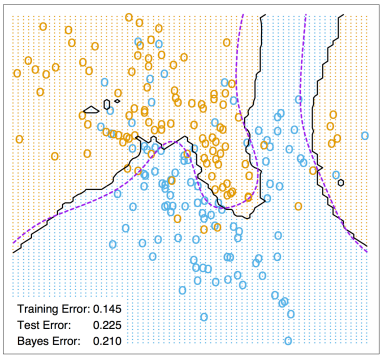
(b) The following pictures, show the output of several different classifiers. Recall that the thick line is the decision boundary determined by the classifier; you can ignore the dashed lines.



(I)



(II)



(III)

For each of the three pictures:

- Name at least one classifier which could have produced this solution. Explain why.
- Name at least one classifier which could not have produced the solution. Explain why not.

Solution:

	(I)	(II)	(III)
could be generated by	logistic regression or LDA or support vector classifier	SVM with kernels	K-nearest-neighbor classifier
reason	linear boundary, class overlap	smooth, non-linear boundary	non-linear boundary
could not be generated by	K-nearest-neighbor classifier	any linear classifier	any linear classifier

Problem 3: Support vector machines(5+5+6+4 points)

Here we consider an alternative formulation of the soft margin SVM. Given some training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ in $\mathbb{R}^{p+1}$(i.e. each $\mathbf{x}_i$ is a p-dimensional vector and y_i is the one dimensional response), in lecture, in order to estimate $\mathbf{v}_H$ ($\mathbf{v}_H = (v_H^1, \ldots, v_H^p) \in \mathbb{R}^p$) and b which determine the decision boundary, we need to solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{v}_H, b, \xi} \quad & \|\mathbf{v}_H\|^2 + C\sum_{i=1}^{n} \xi_i^2 \\ \text{s.t.} \quad & y_i(\langle \mathbf{v}_H, \mathbf{x}_i \rangle - b) \geq 1 - \xi_i, \quad \text{for } i = 1, \ldots, n \\ & \xi_i \geq 0, \quad \text{for } i = 1, \ldots, n. \end{aligned}$$

Alternatively, we can directly minimize the cost:

$$C(\mathbf{v}_H, b) = \sum_{i=1}^{n} \max\left\{0, 1 - y_i\left(\mathbf{v}_H^T \mathbf{x}_i - b\right)\right\} + \frac{\lambda}{2}\|\mathbf{v}_H\|^2, \tag{1}$$

where $\|\mathbf{v}_H\|^2 = \sum_{j=1}^{p} (v_H^j)^2$.

(a) If we formulate SVM in this form, λ will be the tuning parameter of the problem. Will large value of λ result in a larger margin or a smaller one? Explain briefly.

(b) Let $f(\mathbf{x}) = \mathbf{v}_H^T \mathbf{x} - b$. If we treat f as our prediction and define the Hinge loss function to be $L_1(y, f) = \max\{0, 1 - yf\}$, then the optimization problem (1) has the form *loss + penalty*. More explicitly, in (1) the first term corresponds to the empirical risk (loss on the training set) and the second term is the penalty. If we use a different loss function—binomial deviance $L_2(y, f) = \log\left(1 + e^{-yf}\right)$ to replace the Hinge loss in (1), what is the corresponding optimization problem?

Solution:

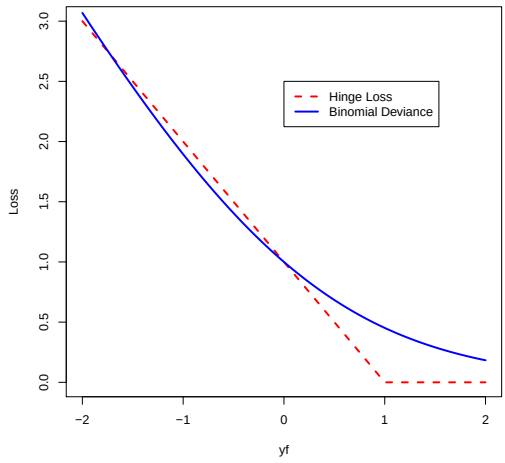
(a) Large value of λ will give us a classifier with a larger margin, since it implies a heavy penalty on $\|\mathbf{v}_H\|$. The solution will prefer a small $\|\mathbf{v}_H\|$, while $1/\|\mathbf{v}_H\|$ is the margin.

(b) The optimization problem is

$$\min_{\mathbf{v}_H, b} \quad \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i(\mathbf{v}_H^T \mathbf{x}_i - b)\right)\right) + \frac{\lambda}{2}\|\mathbf{v}_H\|^2.$$

(c) Consider the optimization problem you derived in previous part. If we want to use gradient descent to solve this problem, write down the gradient update for $\mathbf{v}_H$ and b in each iteration.

(d) The following picture shows the plot of these two loss functions as a function of yf. Binomial deviance can be regarded as a smooth approximation to the Hinge loss. Will you expect the solution to the optimization problem you derived in part (b) to be very different from or similar to the solution to (1)?



Solution:

(c)

$$(v_H^j)^{(r+1)} = (v_H^j)^{(r)} - \left(\lambda (v_H^j)^{(r)} - \sum_{i=1}^{n} \frac{y_i x_{ij} \exp\left(-y_i(\mathbf{x}_i^T \mathbf{v}_H^{(r)} - b^{(r)})\right)}{1 + \exp\left(-y_i(\mathbf{x}_i^T \mathbf{v}_H^{(r)} - b^{(r)})\right)} \right), \quad \text{for } j = 1, \ldots, p,$$

$$b^{(r+1)} = b^{(r)} - \left(\sum_{i=1}^{n} \frac{y_i \exp\left(-y_i(\mathbf{x}_i^T \mathbf{v}_H^{(r)} - b^{(r)})\right)}{1 + \exp\left(-y_i(\mathbf{x}_i^T \mathbf{v}_H^{(r)} - b^{(r)})\right)} \right).$$

(d) The solution sould be similar to that of (1), since binomial deviance can be viewed as a smooth approximation to the Hinge loss. One difference is that if we use binomial deviance, then we cannot define the concept of "support vectors".

Problem 4: Variable selection (2+2+4+2 points)

You are asked to fit a linear regression model to some microarray data. You have a response vector $\mathbf{y}$ with 53 measurements, and a regression matrix $\mathbf{X}$ with 53 rows and 1400 columns (each column represents a gene, for each observation, we measured the expression levels for 1400 genes). You are told that the ultimate goal is to be able to accurately predict values of y, given a measured gene expression pattern.

(a) Do you anticipate any problem with assignment?

(b) You decide to build a model by forward stepwise selection. You start with the constant model, and active set of variables $\mathcal{A}$ is empty. At each step, you want to include the variable into the model that reduces the residual-sum-of-squares the most. That is, you will add that variable to $\mathcal{A}$, and then compute the least squares fit using all the predictors in $\mathcal{A}$. When will your algorithm stop, and after how many steps? (be precise, and give conditions if necessary). Characterize the model at the final step.

(c) This sequence of models can be indexed by model size $k = |\mathcal{A}|$. Describe how you would use 10-fold cross-validation select the optimal model.

(d) Can you suggest some alternative ways to build the model?

Solution:

(a) In the problem, the number of predictors is much larger than the number of observations. If we use a linear model, it will over-fit. Therefore, it is necessary to perform variable selection in this case.

(b) After 53 steps, i.e., when the number of predictors included in the model is equal to the number of observations, the algorithm will stop. At the final step, the RSS of the model will be zero.

(c) Assume that we have a sequence of model $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_p$. To perform 10-fold cross-validation, we randomly divide the data into 10 folds of roughly the same size. For each model $\mathcal{M}_k$, we use one block of data as validation set, and the rest as training set. After fitting the model on the training set, we calculate the averaged loss on the validation set. We repeat this step for 10 times, with each block of data being the validation set at each time. The average test error over these 10 iterations is the cross-validation error. We want to select the optimal model which either minimize the cross-validation error or satisfies the one standard deviation rule.

(d) We can also use the lasso or ridge regression to build the model.

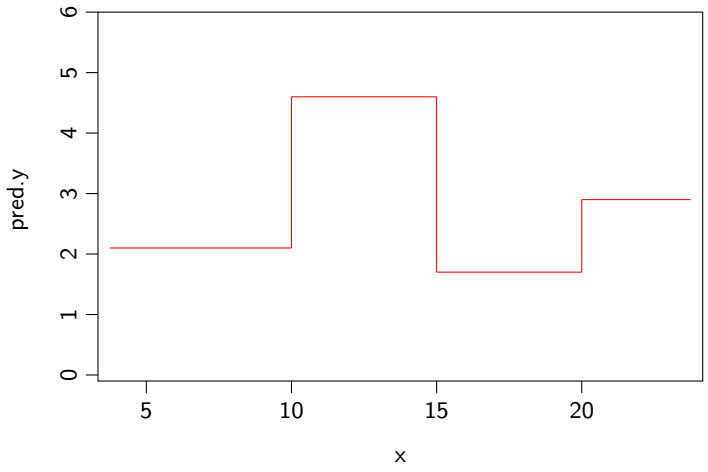
Problem 5: Step function regression(4+6+5+5 points)

We fit a step function regression on a dataset with a single predictor X. Three knots $c_1 = 10, c_2 = 15, c_3 = 20$ in the range of X are selected. We construct 4 variables

$$C_1(X) = \mathbb{1}_{\{X<c_1\}}, \quad C_2(X) = \mathbb{1}_{\{c_1\leq X<c_2\}}, \quad C_3(X) = \mathbb{1}_{\{c_2\leq X<c_3\}}, \quad C_4(X) = \mathbb{1}_{\{c_3\leq X\}},$$

where $\mathbb{1}_{\{\cdot\}}$ is an indicator function. For example, C_1 takes value 1 when $X < c_1$ is true; and is equal to 0 otherwise. Note that the linear model using $C_1(X), C_2(X), C_3(X), C_4(X)$ as predictors is:

$$Y = \beta_1 C_1(X) + \beta_2 C_2(X) + \beta_3 C_3(X) + \beta_4 C_4(X) + \epsilon, \quad \epsilon \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2). \tag{2}$$



Assume that we have a data set which only contains four observations: $(x_1, y_1) = (8, 2.1), (x_2, y_2) = (11, 4.6), (x_3, y_3) = (17, 1.7)$, and $(x_4, y_4) = (21, 2.9)$.

(a) $\mathbf{N}$ is a matrix with each row representing an observation and each column representing a predictor, i.e., the (i, j)th element of $\mathbf{N}$ is $C_j(x_i)$. Write down the matrix $\mathbf{N}$ explicitly.

(b) If we use ordinary least squares to estimate $\beta = (\beta_1, \beta_2, \beta_3, \beta_4)^T$, what is your estimate? Draw the fitted step function in the figure above.

Solution:

(a)

$$\mathbf{N} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(b) Since the regression matrix is the identity matrix, least square estimate is

$$\hat{\beta}_{\text{OLS}} = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{y} = (2.1, 4.6, 1.7, 2.9)^T$$

(c) Now we fit ridge regression using the same set of predictors on this data set:

$$\hat{\beta}_{\text{ridge}} = \underset{\beta}{\text{argmax}}\ (\mathbf{y} - \mathbf{N}\beta)^T(\mathbf{y} - \mathbf{N}\beta) + \lambda\|\beta\|_2^2,$$

where $\mathbf{y} = (y_1, y_2, y_3, y_4)^T$, and $\|\beta\|_2^2 = \sum_{j=1}^4 \beta_j^2$. Using the tuning parameter $\lambda = 3$, what is the ridge estimate of β?

(d) In this part, we fit a lasso regression on this data set, and still use C_1, C_2, C_3, and C_4 as predictors

$$\hat{\beta}_{\text{lasso}} = \underset{\beta}{\text{argmax}}\ \frac{1}{2}(\mathbf{y} - \mathbf{N}\beta)^T(\mathbf{y} - \mathbf{N}\beta) + \lambda\sum_{j=1}^4 |\beta_j|.$$

If $\lambda = 2.5$, what is the lasso estimate?

Solution:

(c) Given the regression matrix being the identity, we know that $\hat{\beta}_j = \frac{y_j}{1+\lambda}$. Therefore,

$$\hat{\beta}_{\text{ridge}} = (0.52, 1.15, 0.425, 0.725)^T.$$

(d) In this special case,

$$\hat{\beta}_j = \begin{cases} y_j - \lambda & \text{if } y_j > \lambda, \\ 0 & \text{if } |y_j| \le \lambda, \\ y_j + \lambda & \text{if } y_j < -\lambda. \end{cases}$$

Then, we have

$$\hat{\beta}_{\text{lasso}} = (0, 2.1, 0, 0.4)^T$$

Problem 6: Principle components analysis (5+5 points)

We have gene expression measurements on 1000 genes for 100 patients with cancer. The expression levels have already been appropriately scaled. Let $\mathbf{X}$ be the 1000 by 100 matrix of data. Each column of data comes from one patient measured on one microarray. Some patients have cancer type A and the others have cancer type B. You want to discover which genes have expression higher or lower in the two kinds of cancer.

You compute the first principle component $\mathbf{v}$ of the observations ($\mathbf{v}$ is a vector of length 100), find that it explains 20% of the variation and note that it is fairly highly correlated (correlation 0.7) with the vector $\mathbf{e}$ consisting of 50 -1s followed by 50 $+1$s.

(a) Describe how principle components are derived and what is meant by the phrase "explains 20% of the variation".

(b) You go back to your collaborator and find that the first 50 measurements came from on production run and the next 50 from a different production run. Given this information, you want to adjust for production batch before analyzing the data. So you decide to regress each gene against $\mathbf{v}$ and take residuals. Then you plan to use the residual data to compare cancer A to cancer B. Is this a good way to adjust for the production batch? Explain your answer.

Solution:

(a) It means if we project the data onto the first principle component, then the variance of the projections constitutes about 20% of the total variance of the data.

(b) No, if we regress with respect to the first principle component and then take the residuals, we may remove some biological effect. A better way is to regress with respect to $\mathbf{e}$.

Problem 7: Numerical optimization (5+5 points)

We have computed maximum likelihood estimators analytically, but since they are defined by a maximization problem, they can also be computed by numerical optimization. Suppose $p(x|\theta)$ is a probability density with parameter space $\mathbb{R}^2$, and our statistical model is

$$M = \left\{ p(x|\theta) \,\middle|\, \theta \in \mathbb{R}^2 \right\} .$$

For observations $x_1, \ldots, x_n$, let L_n be the *negative* log-likelihood, i.e.

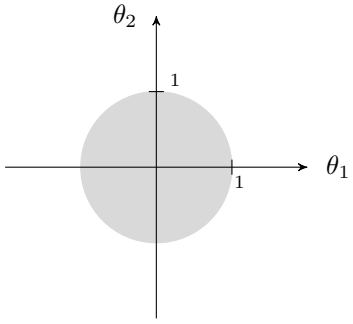
$$L_n(\theta) := -\log \prod_{i=1}^{n} p(x_i|\theta) .$$

We write $\nabla L_n(\theta)$ and $H_{L_n}(\theta)$ for the first two derivatives and assume that L_n is strictly convex (the Hessian $H_{L_n}(\theta)$ is positive definite for all θ).

(a) Name an algorithm that can be used to find the maximum likelihood estimator. (There is more than one possible answer; one is sufficient.) Write down the update step of the algorithm you have chosen, i.e. the equation for computing $\theta^{(m+1)}$ from $\theta^{(m)}$ such that the sequence $\theta^{(1)}, \theta^{(2)}, \ldots$ converges to the maximum likelihood estimator.

(b) Now suppose we restrict our model to those distributions $p(x|\theta)$ for which θ is a point in the gray area marked in the picture. Can you formulate maximum likelihood estimation in this model as a constrained optimization problem?

(You do *not* have to apply any optimization, only write down the problem.)



Solution:

(a) Possible answers: Gradient descent, Newton.

Step for gradient descent:

$$\theta^{(m+1)} := \theta^{(m)} - \nabla L_n(\theta^{(m)})$$

Step for Newton:

$$\theta^{(m+1)} := \theta^{(m)} - H_{L_n}(\theta^{(m)})^{-1} \cdot \nabla L_n(\theta^{(m)})$$

(b) The gray, constrained area of parameter space consists of those θ with $\sqrt{\theta_1^2 + \theta_2^2} \leq 1$. Constrained maximum likelihood problem:

$$\begin{aligned} & \min L_n \\ \text{s.t.} \quad & \|\theta\|_2 - 1 \leq 0 \end{aligned}$$