CS 189/289A Introduction to Machine Learning
Spring 2021 Jonathan Shewchuk Midterm

- The exam is open book, open notes for material on **paper**. On your computer screen, you may have only this exam, Zoom (if you are running it on your computer instead of a mobile device), and four browser windows/tabs: Gradescope, the exam instructions, clarifications on Piazza, and the form for submitting clarification requests.
- You will submit your answers to the multiple-choice questions directly into Gradescope via the assignment "**Midterm – Multiple Choice**"; please **do not** submit your multiple-choice answers on paper. If you are in the DSP program and have been granted extra time, select the "DSP, 150%" or "DSP, 200%" option. By contrast, you will submit your answers to the written questions by writing them on paper by hand, scanning them, and submitting them through Gradescope via the assignment "**Midterm – Free Response**."
- Please write your name at the top of each page of your written answers. (You may do this before the exam.) **Please start each top-level question (Q2, Q3, etc.) on a new sheet of paper. Clearly label all written questions and all subparts of each written question**.
- You have **80 minutes to complete the midterm exam (7:40–9:00 PM)**. (If you are in the DSP program and have an allowance of 150% or 200% time, that comes to 120 minutes or 160 minutes, respectively.)
- When the exam ends (9:00 PM), **stop writing**. You must submit your multiple-choice answers before 9:00 PM sharp. **Late multiple-choice submissions will be penalized at a rate of 5 points per minute after 9:00 PM.** (The multiple-choice questions are worth 40 points total.)
- From 9:00 PM, you have 15 minutes to scan the written portion of your exam and turn it into Gradescope via the assignment "Midterm – Free Response." Most of you will use your cellphone/pad and a third-party scanning app. If you have a physical scanner, you may use that. **Late written submissions will be penalized at a rate of 10 points per minute after 9:15 PM.** (The written portion is worth 60 points total.)
- Following the exam, you must use Gradescope's **page selection mechanism** to mark which questions are on which pages of your exam (as you do for the homeworks). Please get this done before 2:00 AM. This can be done on a computer different than the device you submitted with.
- The total number of points is 100. There are 10 multiple choice questions worth 4 points each, and four written questions worth a total of 60 points.
- For multiple answer questions, fill in the bubbles for **ALL correct choices:** there may be more than one correct choice, but there is always at least one correct choice. **NO partial credit** on multiple answer questions: the set of all correct answers must be checked.

Q1. [40 pts] Multiple Answer

Fill in the bubbles for **ALL correct choices**: there may be more than one correct choice, but there is always at least one correct choice. **NO partial credit**: the set of all correct answers must be checked.

(a) [4 pts] Which of the following cost functions are smooth—i.e., having continuous gradients everywhere?

- ○ A: the perceptron risk function
- ● B: the sum (over sample points) of logistic losses
- ● C: least squares with ℓ_2 regularization
- ○ D: least squares with ℓ_1 regularization

(b) [4 pts] Which of the following changes would commonly cause an SVM's margin $1/\|w\|$ to shrink?

- ● A: Soft margin SVM: increasing the value of C
- ● B: Hard margin SVM: adding a sample point that violates the margin
- ○ C: Soft margin SVM: decreasing the value of C
- ○ D: Hard margin SVM: adding a new feature to each sample point

The greater the value of C is, the higher the penalty for violating the margin. The soft margin shrinks to compensate.

If you add a sample point that violates the margin, a hard margin always shrinks.

If you add a feature, the old solution can still be used (by setting the weight associated with the new feature to zero). Although the new feature might enable a new solution with a wider margin, the optimal solution can't be worse than the old solution.

(c) [4 pts] Recall the logistic function $s(\gamma)$ and its derivative $s'(\gamma) = \frac{\mathrm{d}}{\mathrm{d}\gamma} s(\gamma)$. Let γ^* be the value of γ that maximizes $s'(\gamma)$.

- ○ A: $\gamma^* = 0.25$
- ● B: $s(\gamma^*) = 0.5$
- ○ C: $s'(\gamma^*) = 0.5$
- ● D: $s'(\gamma^*) = 0.25$

A glance at the logistic curve and its symmetry makes it clear that the slope is maximized when $\gamma^* = 0$ and $s(\gamma^*) = 0.5$. Recall that $s' = s(1 - s)$; hence $s'(\gamma^*) = 0.25$.

(d) [4 pts] You are running logistic regression to classify two-dimensional sample points $X_i \in \mathbb{R}^2$ into two classes $y_i \in \{0, 1\}$ with the regression function $h(z) = s(w^\top z + \alpha)$, where s is the logistic function. Unfortunately, regular logistic regression isn't fitting the data very well. To remedy this, you try appending an extra feature, $\|X_i\|^2$, to the end of each sample point X_i. After you run logistic regression again with the new feature, the decision boundary in $\mathbb{R}^2$ could be

- ● A: a line.
- ● B: a circle.
- ○ C: an ellipse.
- ○ D: an S-shaped logistic curve.

The decision boundary is the set of points $\{z : s(w^\top z + \alpha) = 0.5\}$, which is the set of points $\{z : w^\top z + \alpha = 0\}$. If $z = [x_1 \quad x_2 \quad x_1^2 + x_2^2]$, then the decision boundary for points in x-space is $w_1 x_1 + w_2 x_2 + w_3(x_1^2 + x_2^2) + \alpha = 0$. This equation can express arbitrary circles and lines. It can't express ellipses because there is only one quadratic feature, and it causes x_1^2 and x_2^2 to always have the same coefficient. It definitely can't express any S-shaped curves.

(e) [4 pts] We are performing least-squares linear regression, with the use of a fictitious dimension (so the regression function isn't restricted to satisfy $h(0) = 0$). Which of the following will never increase the training error, as measured by the mean squared-error cost function?

- ● A: Adding polynomial features
- ○ B: Using backward stepwise selection to remove some features, thereby reducing validation error
- ○ C: Using Lasso to encourage sparse weights
- ● D: Centering the sample points

Adding new features never increases the training error. Centering the sample points never changes the training error (as we're using a fictitious dimension), as the change in coordinate system can be compensated for by a translation of the regression function (effected by a change in the bias term). Options B and C usually increase training error for the purpose of reducing validation error.

(f) [4 pts] Given a design matrix $X \in \mathbb{R}^{n \times d}$, labels $y \in \mathbb{R}^n$, and $\lambda > 0$, we find the weight vector w^* that minimizes $||Xw - y||^2 + \lambda||w||^2$. Suppose that $w^* \neq 0$.

- ● A: The variance of the method decreases if λ increases enough.
- ○ B: There may be multiple solutions for w^*.
- ● C: The bias of the method increases if λ increases enough.
- ○ D: $w^* = X^+ y$, where X^+ is the pseudoinverse of X.

A and C: as $\lambda \to \infty$, $w^* \to 0$ so the variance also approaches zero and the bias grows. When $\lambda > 0$, the objective is strongly convex and the minimizer is unique. D holds only when $\lambda = 0$, which we have said is not the case.

(g) [4 pts] **The following two questions use the following assumptions.** You want to train a dog identifier with Gaussian discriminant analysis. Your classifier takes an image vector as its input and outputs 1 if it thinks it is a dog, and 0 otherwise. You use the CIFAR10 dataset, modified so all the classes that are not "dog" have the label 0. Your training set has 5,000 dog images and 45,000 non-dog ("other") images. Which of the following statements seem likely to be correct?

○ A: LDA has an advantage over QDA because the two classes have different numbers of training examples.

○ B: QDA has an advantage over LDA because the two classes have different numbers of training examples.

○ C: LDA has an advantage over QDA because the two classes are expected to have very different covariance matrices.

● D: QDA has an advantage over LDA because the two classes are expected to have very different covariance matrices.

The "dog" class (label 1) only contains images of dogs, so compared to the "other" class (label 0), it should have less variance. The "other" class has images from many different classes, so it will have greater variance. QDA fits different covariance matrices to different classes, so it is more appropriate than LDA, which fits a single covariance matrix to both classes.

(h) [4 pts] **This question is a continuation of the previous question.** You train your classifier with LDA and the 0-1 loss. You observe that at test time, your classifier always predicts "other" and never predicts "dog." What is a likely reason for this and how can we solve it? (Check all that apply.)

● A: Reason: The prior for the "other" class is very large, so predicting "other" on every test point minimizes the (estimated) risk.

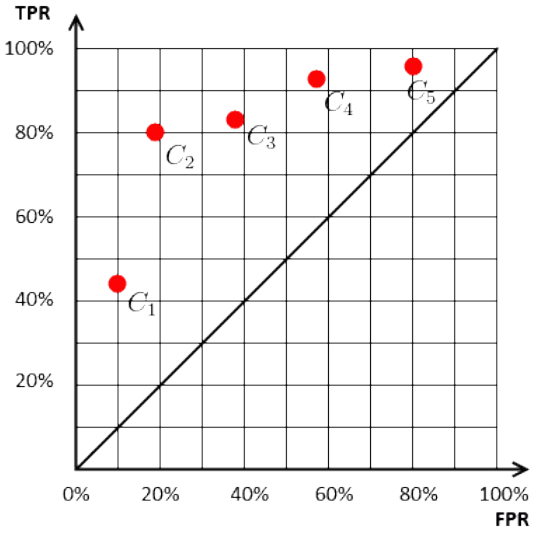
○ B: Reason: As LDA fits the same covariance matrix to both classes, the class with more examples will be predicted for all points in $\mathbb{R}^d$.

● C: Solve it by using a loss function that penalizes dogs misclassified as "other" more than "others" misclassified as dogs.

○ D: Solve it by learning an isotropic pooled covariance instead of an anisotropic one; that is, the covariance matrix computed by LDA has the form $\sigma^2 I$.

A: The 0-1 loss does not account for the imbalance in the number of examples per class, so it is minimized by predicting the class with the larger prior for a very large set of inputs if that prior is large enough (in this case, that set is large enough to contain the entire set of training images).
B: LDA can still predict the class with fewer examples when trained to minimize the 0-1 loss if the means of the two classes are different enough.
C: An asymmetric loss can counteract the larger prior on the "other" class.
D: lolno

(i) [4 pts] We do an ROC analysis of 5 binary classifiers C_1, C_2, C_3, C_4, C_5 trained on the training points X_{train} and labels y_{train}. We compute their true positive and false positive rates on the validation points X_{val} and labels y_{val} and plot them in the ROC space, illustrated below. In X_{val} and y_{val}, there are n_p points in class "positive" and n_n points in class "negative." We use a 0-1 loss.



ROC analysis of five classifiers. FPR = false positive rate; TPR = true positive rate.

● A: If $n_p = n_n$, C_2 is the classifier with the highest validation accuracy.

● B: If $n_p = n_n$, all five classifiers have higher validation accuracy than any random classifier.

● C: There exists some n_p and n_n such that C_1 is the classifier with the highest validation accuracy.

○ D: There exists some n_p and n_n such that C_3 is the classifier with the highest validation accuracy.

If $n_p = n_n$, the validation accuracy of each classifier is 0.5(True Positive Rate) +0.5(True Negative Rate) and C_2 is the best classifier, with an accuracy of about 80%. All five classifiers lie above the diagonal line, so they all have validation accuracy better than 50%, whereas any random classifier (biased or not) will have a validation accuracy of exactly 50% when $n_p = n_n$. C_1 is the best classifier when nearly all the validation data is negative, but C_3 is never the best classifier for any n_p/n_n, as it lies below the ROC convex hull.

(j) [4 pts] Tell us about feature subset selection.

○ A: Ridge regression is more effective for feature subset selection than Lasso.

○ B: If the best model uses only features 2 and 4 (i.e., the second and fourth columns of the design matrix), forward stepwise selection is guaranteed to find that model.

○ C: Stepwise subset selection uses the accuracy on the training data to decide which features to include.

● D: Backward stepwise selection could train a model with only features 1 and 3. It could train a model with only features 2 and 4. But it will never train both models.

A: Lasso can be used for subset selection because it encourages weights of zero. Ridge regression almost never produces weights of exactly zero.

C: Subset selection uses validation accuracy, not training accuracy.

B: Stepwise subset selection can miss the best subset. If the best 1-feature model uses only feature 3, then forward stepwise selection won't investigate the model that uses features 2 and 4.

D: It will never train both because it only tries training 2-feature models when it has already reduced the number of features to three, and those three features cannot contain all of features 1, 2, 3, and 4.

Q2. [14 pts] Eigendecompositions

(a) [5 pts] Consider a symmetric, square, real matrix $A \in \mathbb{R}^{d \times d}$. Let $A = V \Lambda V^\top$ be its eigendecomposition. Let v_i denote the ith column of V. Let λ_i denote Λ_{ii}, the scalar component on the ith row and ith column of Λ.

Consider the matrix $M = \alpha A - A^2$, where $\alpha \in \mathbb{R}$. What are the eigenvalues and eigenvectors of M? (Expressed in terms of parts of A's eigendecomposition and α. No proof required.)

Every eigenvector v_i of A is also an eigenvector of M. For each eigenvalue λ_i of A, $\alpha\lambda_i - \lambda_i^2$ is an eigenvalue of M.

(b) [4 pts] Suppose that A is a sample covariance matrix for a set of n sample points stored in a design matrix $X \in \mathbb{R}^{n \times d}$, and that $\alpha \in \mathbb{R}$ is a fixed constant. Is it always true (for any such A and α) that there exists another design matrix $Z \in \mathbb{R}^{n \times d}$ such that $M = \alpha A - A^2$ is the sample covariance matrix for Z? Explain your answer.

No. M could have negative eigenvalues, but all sample covariance matrices are positive semidefinite.

(c) [5 pts] In lecture, we talked about decorrelating a centered design matrix $\dot{X}$. We used an eigendecomposition to do that. Explain (in English, not math) what the eigendecomposition tells us about the sample points, and how that information helps us decorrelate a design matrix.

The eigenvectors of _____ tell us
_____.

With this information, we decorrelate the centered design matrix by
_____.

The eigenvectors of the sample covariance matrix tell us some orthogonal directions (alternative coordinate axes) along which the points are not correlated.

With this information, we decorrelate the centered design matrix by doing a change of coordinates (a rotation) to a coordinate system in which the features have covariance zero.

Q3. [10 pts] Maximum Likelihood Estimation

There are 5 balls in a bag. Each ball is either red or blue. Let θ (an integer) be the number of blue balls. We want to estimate θ, so we draw 4 balls **with replacement** out of the bag, replacing each one before drawing the next. We get "blue," "red," "blue," and "blue" (in that order).

(a) [5 pts] Assuming θ is fixed, what is the likelihood of getting exactly that sequence of colors (expressed as a function of θ)?

$\mathcal{L}(\theta; X) = P(X; \theta) = \left(\frac{5-\theta}{5}\right)\left(\frac{\theta}{5}\right)^3$.

(b) [3 pts] Draw a table showing (as a fraction) the likelihood of getting exactly that sequence of colors, for every value of θ from zero to 5 inclusive.

θ	$\mathcal{L}(\theta; \langle$ blue, red, blue, blue $\rangle)$
0	?
1	?
2	?
3	?
4	?
5	?

θ	$\mathcal{L}(\theta; \langle$ blue, red, blue, blue $\rangle)$
0	0
1	4 / 625
2	24 / 625
3	54 / 625
4	64 / 625
5	0

(c) [2 pts] What is the maximum likelihood estimate for θ? (Chosen among all integers; not among all real numbers.)

The maximum likelihood estimate for θ is 4.

Q4. [20 pts] Tikhonov Regularization

Let's take a look at a more complicated version of ridge regression called *Tikhonov regularization*. We use a regularization parameter similar to λ, but instead of a scalar, we use a real, square matrix $\Gamma \in \mathbb{R}^{d \times d}$ (called the *Tikhonov matrix*). Given a design matrix $X \in \mathbb{R}^{n \times d}$ and a vector of labels $y \in \mathbb{R}^n$, our regression algorithm finds the weight vector $w^* \in \mathbb{R}^d$ that minimizes the cost function

$$J(w) = \|Xw - y\|_2^2 + \|\Gamma w\|_2^2.$$

(a) [7 pts] Derive the normal equations for this minimization problem—that is, a linear system of equations whose solution(s) is the optimal weight vector w^*. **Show your work.** (If you prefer, you can write an explicit closed formula for w^*.)

The gradient of the cost function is

$$\nabla J(w) = 2X^\top(Xw - y) + 2\Gamma^\top \Gamma w.$$

Setting $\nabla J = 0$ gives the normal equations,

$$(X^\top X + \Gamma^\top \Gamma) w^* = X^\top y.$$

If the matrix is not singular, we can write the closed formula

$$w^* = (X^\top X + \Gamma^\top \Gamma)^{-1} X^\top y.$$

(b) [3 pts] Give a simple, sufficient and necessary condition on Γ (involving *only* Γ; not X nor y) that guarantees that $J(w)$ has only one unique minimum w^*. (To be precise, the uniqueness guarantee must hold for *all* values of X and y, although the unique w^* will be different for different values of X and y.) (A sufficient but not necessary condition will receive part marks.)

The following conditions are all equivalent to each other, and are all correct answers. $\Gamma^\top \Gamma$ is positive definite. Γ is invertible. Γ is nonsingular. Γ has full rank. Γ has rank d.

(c) [5 pts] Recall the Bayesian justification of ridge regression. We impose an isotropic normal prior distribution on the weight vector—that is, we assume that $w \sim \mathcal{N}(0, \sigma^2 I)$. (This encodes our suspicion that small weights are more likely to be correct than large ones.) Bayes' Theorem gives us a posterior distribution $f(w|X, y)$. We apply maximum likelihood estimation (MLE) to estimate w in that posterior distribution, and it tells us to find w by minimizing $\|Xw - y\|_2^2 + \lambda \|w\|_2^2$ for some constant λ.

Suppose we change the prior distribution to an **anisotropic** normal distribution: $w \sim \mathcal{N}(0, \Sigma)$ for some symmetric, positive definite covariance matrix Σ. Then MLE on the new posterior tells us to do Tikhonov regularization! What value of Γ does MLE tells us to use when we minimize $J(w)$?

Give a one-sentence explanation of your answer.

$\Gamma = \Sigma^{-1/2}$. (Answers that incorporate an extra scalar factor, like $\Gamma = \lambda \Sigma^{-1/2}$, are also acceptable—maybe even preferable.) Because the log posterior will include a term equal to some constant times $w^\top \Sigma^{-1} w$, which is equal to $\|\Gamma w\|^2$ if $\Gamma = \Sigma^{-1/2}$.

(d) [5 pts] Suppose you solve a Tikhonov regularization problem in a two-dimensional feature space ($d = 2$) and obtain a weight vector w^* that minimizes $J(w)$. The solution w^* lies on an isocontour of $\|Xw - y\|_2^2$ and on an isocontour of $\|\Gamma w\|_2^2$. Draw a diagram that plausibly depicts both of these two isocontours, in a case where Γ is **not** diagonal and $y \neq 0$. (You don't need to choose specific values of X, y, or Γ; your diagram just needs to look plausible.)

Your diagram must contain the following elements:

- The two axes (coordinate system) of the space you are optimizing in, with both axes labeled.
- The specified isocontour of $\|Xw - y\|^2$, labeled.
- The specified isocontour of $\|\Gamma w\|^2$, labeled.
- The point w^*.

These elements must be in a plausible geometric relationship to each other.

The diagram should contain: the two coordinate axes labeled w_1 and w_2; an ellipse or circle, **not** centered at the origin, labeled $\|Xw - y\|^2$; an ellipse, **not** axis-aligned and **not** circular, centered at the origin, labeled $\|\Gamma w\|^2$; a point w^* on both ellipses; and the two ellipses must touch each other tangentially without crossing each other.

Q5. [16 pts] Multiclass Bayes Decision Theory

Let's apply Bayes decision theory to three-class classification. Consider a weather station that constantly receives data from its radar systems and must predict what the weather will be on the next day. Concretely:

- The input X is a scalar value representing the level of cloud cover, with only four discrete levels: 25, 50, 75, and 100 (the percentage of cloud cover).
- The station must predict one of three classes Y corresponding to the weather tomorrow. $Y = y_0$ means sunny, y_1 means cloudy, and y_2 means rain.
- The priors for each class are as follows: $P(Y = y_0) = 0.5$, $P(Y = y_1) = 0.3$, and $P(Y = y_2) = 0.2$.
- The station has measured the cloud cover on the days prior to 100 sunny days, 100 cloudy days, and 100 rainy days. From these numbers they estimated the class-conditional probability mass functions $P(X|Y)$:

Prior-Day Cloud Cover (X)	Sunny, $P(X\|Y = y_0)$	Cloudy, $P(X\|Y = y_1)$	Rain, $P(X\|Y = y_2)$
25	0.7	0.3	0.1
50	0.2	0.3	0.1
75	0.1	0.3	0.3
100	0	0.1	0.5

- We use an asymmetric loss. Let z be the predicted class and y the true class (label).

$$L(z, y) = \begin{cases} 0 & z = y, \\ 1 & y = y_0 \text{ and } z \neq y_0, \\ 2 & y = y_1 \text{ and } z \neq y_1, \\ 4 & y = y_2 \text{ and } z \neq y_2. \end{cases}$$

(a) [8 pts] Consider the constant decision rule $r_0(x) = y_0$, which *always* predicts y_0 (sunny). What is the risk $R(r_0)$ of the decision rule r_0? Your answer should be a number, but **show all your work**.

$$\begin{aligned} R(r_0(x)) &= \sum_i \sum_x P(Y = y_i) L(r_0(x), y_i) P(X = x|Y = y_i) \\ &= P(Y = y_1) L(y_0, y_1) \sum_x P(X = x|Y = y_1) + P(Y = y_2) L(y_0, y_2) \sum_x P(X = x|Y = y_2) \\ &= 0.3 \cdot 2 \cdot 1 + 0.2 \cdot 4 \cdot 1 \\ &= 1.4. \end{aligned}$$

As r_0 does not actually vary with x at all, it is also acceptable to notice that we don't have to break down the risk by x:

$$\begin{aligned} R(r_0(x)) &= R(y_0) \\ &= \sum_i P(Y = y_i) L(y_0, y_i) \\ &= 0.3 \cdot 2 + 0.2 \cdot 4 \\ &= 1.4. \end{aligned}$$

(b) [8 pts] Derive the Bayes optimal decision rule $r^*(x)$—the rule that minimizes the risk $R(r^*)$.

Hint: Write down a table calculating $L(z, y_i)\, P(X|Y = y_i)\, P(Y = y_i)$, for each class y_i and each possible value of X (12 entries total), in the cases where the prediction z is wrong. Then figure out how to use it to minimize R. This problem can be solved without wasting time computing $P(X)$.

The usual principle is to "pick the class with the biggest posterior probability," but in this case we also need to weight each class with an asymmetrical loss. We don't need to fully compute each posterior probability, because we don't need to know the value of the denominator $P(X)$. The following table shows $L(z, y_i)\, P(X|Y = y_i)\, P(Y = y_i)$, for each class y_i and each possible value of X, in the cases where the prediction z is wrong. (When z is correct, the loss is always zero.)

Prior-Day Cloud Cover (X)	Sunny (y_0)	Cloudy (y_1)	Rain (y_2)
25	**0.35**	0.18	0.08
50	0.10	**0.18**	0.08
75	0.05	0.18	**0.24**
100	0.00	0.06	**0.40**

The Bayes optimal decision rule always picks the highest loss-weighted posterior to minimize risk, so we pick the boldfaced classes (table 2) for each of the corresponding inputs X. Formally,

$$r(x) = \begin{cases} y_0 & x = 25, \\ y_1 & x = 50, \\ y_2 & x = 75 \text{ or } x = 100. \end{cases}$$