# CS 189 Spring 2019 Introduction to Machine Learning Midterm

- Please do not open the exam before you are instructed to do so.
- The exam is closed book, closed notes except your one-page cheat sheet.
- **Electronic devices are forbidden on your person**, including cell phones, iPods, headphones, and laptops. Turn your cell phone off and **leave all electronics at the front of the room**, or **risk getting a zero** on the exam.
- You have 1 hour and 20 minutes.
- Please write your initials at the top right of each page after this one (e.g., write "JS" if you are Jonathan Shewchuk). Finish this by the end of your 1 hour and 20 minutes.
- Mark your answers on the exam itself in the space provided. Do **not** attach any extra sheets.
- The total number of points is 100. There are 20 multiple choice questions worth 3 points each, and 4 written questions worth a total of 40 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.

| | |
|---|---|
| First name | |
| Last name | |
| SID | |
| First and last name of student to your left | |
| First and last name of student to your right | |

# Q1. [60 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)** [3 pts] Let $A$ be a real, symmetric $n \times n$ matrix. Which of the following are true about $A$'s eigenvectors and eigenvalues?

● $A$ can have no more than $n$ distinct eigenvalues

○ The vector $\vec{0}$ is an eigenvector, because $A\vec{0} = \lambda\vec{0}$

○ $A$ can have no more than $2n$ distinct unit-length eigenvectors

● We can find $n$ mutually orthogonal eigenvectors of $A$

There can be infinitely many unit-length eigenvectors if the multiplicity of any eigenvector is greater than 1 (so the eiganspace is a plane, and you can pick any vector on the unit circle on that plane).

The 0 vector is not an eigenvector by definition.

**(b)** [3 pts] The matrix that has eigenvector $[1, 2]^\top$ with eigenvalue 2 and eigenvector $[-2, 1]^\top$ with eigenvalue 1 (note that these are not unit eigenvectors!) is

○ $\begin{bmatrix} 9 & -2 \\ -2 & 6 \end{bmatrix}$

○ $\begin{bmatrix} 6 & 2 \\ 2 & 9 \end{bmatrix}$

○ $\begin{bmatrix} 9/5 & -2/5 \\ -2/5 & 6/5 \end{bmatrix}$

● $\begin{bmatrix} 6/5 & 2/5 \\ 2/5 & 9/5 \end{bmatrix}$

**(c)** [3 pts] Consider a **binary classification** problem where we know both of the class conditional distributions exactly. To compute the risk,

○ we need to know all the sample points

● we need to know the class prior probabilities

● we need to know the loss function

○ we need to use gradient descent

**(d)** [3 pts] Assuming we can find algorithms to minimize them, which of the following cost functions will encourage **sparse solutions** (i.e., solutions where many components of $w$ are zero)?

● $\|Xw - y\|_2^2 + \lambda\|w\|_1$

● $\|Xw - y\|_2^2 + \lambda \cdot$ (# of nonzero components of $w$)

● $\|Xw - y\|_2^2 + \lambda\|w\|_1^2$

○ $\|Xw - y\|_2^2 + \lambda\|w\|_2^2$

The first answer is Lasso, which we know finds sparse solutions. The second is Lasso with the penalty squared. Squaring this will leave the same isocontours and this will keep the same properties as Lasso. The third cost function penalizes solutions that are not sparse and will naturally encourage sparse solutions. The last solution is ridge regression, which shrinks weights but does not set weights to zero.

**(e)** [3 pts] Which of the following statements about **logistic regression** are correct?

● The cost function of logistic regression is convex

○ The cost function of logistic regression is concave

○ Logistic regression uses the squared error as the loss function

○ Logistic regression assumes that each class's points are generated from a Gaussian distribution

**(f)** [3 pts] Which of the following statements about stochastic gradient descent and Newton's method are correct?

● Newton's method often converges faster than stochastic gradient descent, especially when the dimension is small

● Newton's method converges in one iteration when the cost function is exactly quadratic with one unique minimum.

○ If the function is continuous with continuous derivatives, Newton's method always finds a local minimum

○ Stochastic gradient descent reduces the cost function at every iteration.

**(g)** [3 pts] Let $X \in \mathbb{R}^{n \times d}$ be a design matrix containing $n$ sample points with $d$ features each. Let $y \in \mathbb{R}^n$ be the corresponding real-valued labels. What is always true about **every** solution $w^*$ that **locally** minimizes the **linear least squares** objective function $\|Xw - y\|_2^2$, no matter what the value of $X$ is?

○ $w^* = X^+ y$ (where $X^+$ is the pseudoinverse)

● $w^*$ satisfies the normal equations

○ $w^*$ is in the null space of $X$

● All of the local minima are global minima

Top left: $w^* = X^+ y$ is the least squares solution with **least norm**. If the null-space of $X$ is non-empty, there are infinitely many other solutions that minimize $\|Xw - y\|_2^2$ but that have larger norm.

Bottom left: If $w^*$ was in the null space of $X$, then $Xw^* = 0$. This can only be a solution to the linear least squares objective if and only if $y$ is also in the null-space of $X$.

Top right: The normal equations $A^T Aw = A^T y$ define all values of $w$ that make zero the gradient of the least squares objective. Therefore any minimizer must satisfy them.

Bottom right: The objective is convex, and therefore all local minimizers are also global minimizers.

**(h)** [3 pts] We are using **linear discriminant analysis** to classify points $x \in \mathbb{R}^d$ into **three** different classes. Let $S$ be the set of points in $\mathbb{R}^d$ that our trained model classifies as belonging to the first class. Which of the following are true?

○ The decision boundary of $S$ is always a hyperplane

● $S$ can be the whole space $\mathbb{R}^d$

● The decision boundary of $S$ is always a subset of a union of hyperplanes

● $S$ is always connected (that is, every pair of points in $S$ is connected by a path in $S$)

Top left: Given that we have three classes, $S$ is defined by two linear inequalities, and therefore its boundary may not be a hyperplane.

Bottom left: Given that $S$ is defined as the points satisfying a set of inequalities, its boundary is a subset of the hyperplanes defined by each of the linear inequalities.

Top right: If the prior for the first class is high enough, the probability of that class could be higher everywhere, and hence $S$ would be the whole space. For example, take $\mu_1 = \mu_2 = \mu_3$ and $\pi_1 > \pi_2 = \pi_3$.

Bottom right: $S$ is a convex polytope defined by the intersection of half-spaces (i.e. the points satisfying a set of linear inequalities). This is a convex set, and therefore it is connected.

**(i)** [3 pts] Which of the following apply to **linear discriminant analysis**?

● You calculate the sample mean for each class

● It approximates the Bayes decision rule

○ You calculate the sample covariance matrix using the mean of all the data points

○ The model produced by LDA is never the same as the model produced by QDA

Top left: Calculating the sample mean within each class is part of LDA by definition

Bottom left: You calculate the sample covariance using the mean for each class, not the mean of all the data points

Top right: LDA finds what the Bayes decision rule would be under the assumption the class conditionals have normal distributions, parameterized by the sample means and covariance

Bottom right: QDA can produce the same covariance for each class as LDA

**(j)** [3 pts] Which of the following are reasons why you might adjust your model in ways that increase the bias?

○ You observe high training error and high validation error

● You observe low training error and high validation error

● You have few data points

○ Your data are not linearly separable

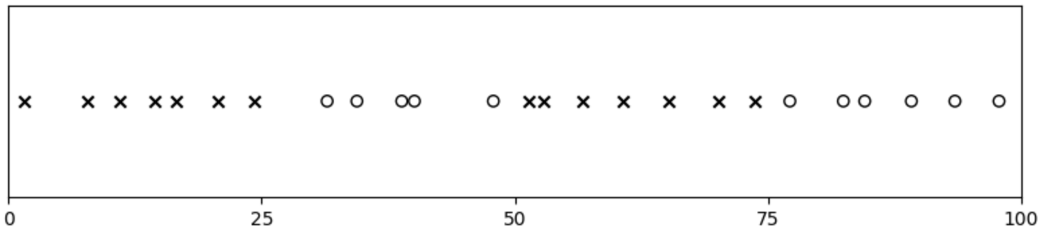
Top left: High training and validation error is a sign of underfitting; higher bias leads to greater underfitting hence you would not want the bias to increase

Bottom left: With few data points, noise in the data has a larger effect on the model. Methods to reduce overfitting to the noise would increase the bias

Top right: Low training error and high validation error is a sign of overfitting; methods to decrease overfitting increase bias

Bottom right: If the data are not linearly separable, you need a more complex decision boundary. Higher bias would reduce the complexity of the decision boundary.

**(k)** [3 pts] Suppose you are given the one-dimensional data $\{x_1, x_2, \ldots, x_{25}\}$ illustrated below and you have only a **hard-margin support vector machine** (with a fictitious dimension) at your disposal. Which of the following modifications can give you 100% training accuracy?



○ Centering the data

○ Add a feature $x_i^2$

● Add a feature that is 1 if $x \leq 50$, or $-1$ if $x > 50$

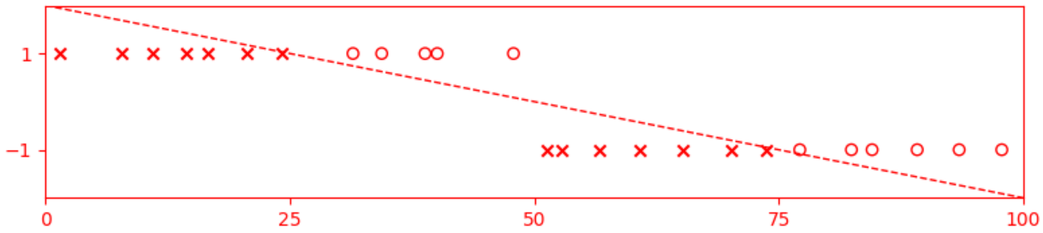
● Add two features, $x_i^2$ and $x_i^3$

○ The performance of SVM is shift invariant, so centering the data won't affect the result;

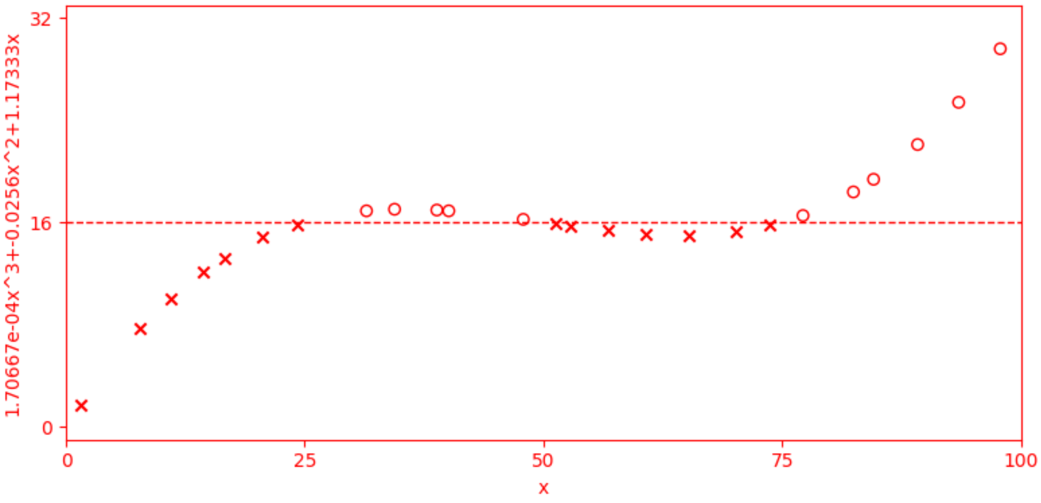
○ A line can separate a quadratic function into at most 3 segments and is not sufficient;

● See image below;

● A line can separate a cubic function into 4 segments. See image below.



Adding "1 if $x_i \leq 50$..." feature



Adding $x_i^2$ and $x_i^3$

**(l)** [3 pts] You are performing **least-squares polynomial regression**. As the degree of your polynomials increases, which of the following is commonly seen to go down at first but then go up?

○ Training error

● Validation error

○ Variance

○ Bias

**(m)** [3 pts] Let $f : \mathbb{R} \to \mathbb{R}$ be a continuous, smooth function whose derivative $f'(x)$ is also continuous. Suppose $f$ has a unique global minimum $x^* \in (-\infty, \infty)$, and you are using **gradient descent** to find $x^*$. You fix some $x^{(0)} \in \mathbb{R}$ and $\epsilon > 0$, and run $x^{(t)} = x^{(t-1)} - \epsilon f'(x^{(t-1)})$ repeatedly. Which of the following statements are true?

○ Gradient descent is sure to converge, to *some* value, for any step size $\epsilon > 0$

○ Assuming gradient descent converges, it converges to $x^*$ if and only if $f$ is convex

● If $f$ has a local minimum $x'$ different from the global one, i.e., $x' \neq x^*$, and $x^{(t)} = x'$ for some $t$, gradient descent will not converge to $x^*$

● If, additionally, $f$ is the objective function of logistic regression, and gradient descent converges, then it converges to $x^*$

The top-left option is false because for a large enough step size, gradient descent may not converge. The bottom-left option is correct because $f'(x') = 0$, so gradient descent will never move from a local minimum. The top-right option is false because you could "accidentally" initialize GD at $x^*$ even if $f$ is non-convex. The bottom-right option is correct because the objective of logistic regression is convex.

**(n)** [3 pts] Suppose you are trying to choose a good subset of features for a **least-squares linear regression** model. Let algorithm A be forward stepwise selection, where we start with zero features and at each step add the new feature that most decreases validation error, stopping only when validation error starts increasing. Let algorithm B be similar, but at each step we include the new feature that most decreases **training** error (measured by the usual cost function, mean squared error), stopping only when training error starts increasing. Which of the following is true?

○ Algorithm B will select no more features than Algorithm A does

○ The first feature chosen by the two algorithms will be the same

● Algorithm B will select at least as many features as Algorithm A does

○ Algorithm A sometimes selects features that Algorithm B does not select

Algorithm B will always select all features, since training error cannot decrease with the addition of a new feature in a linear regression model. There is no guarantee that the first feature chosen will be the same.

**(o)** [3 pts] Suppose you have a **multivariate normal distribution** with a positive definite covariance matrix $\Sigma$. Consider a second multivariate Gaussian distribution whose covariance matrix is $\kappa\Sigma$, where $\kappa = \cos\theta > 0$. Which of the following statements are true about the ellipsoidal isocontours of the second distribution, compared to the first distribution?

○ The principal axes of the ellipsoids would be rotated by $\theta$

○ The principal axes (radii) of the ellipsoids will be scaled by $1/\kappa$

○ The principal axes (radii) of the ellipsoids will be scaled by $\kappa$

● The principal axes (radii) of the ellipsoids will be scaled by $\sqrt{\kappa}$

Multiplying $\Sigma$ by $\kappa$ multiplies the eigenvalues by $\kappa$. The axes of the ellipsoids are scaled by the square roots of the eigenvalues of $\Sigma$.

**(p)** [3 pts] Suppose $M$ and $N$ are **positive semidefinite** matrices. Under what conditions is $M - N$ certain to be positive semidefinite?

○ Never

○ If $M$ and $N$ share all the same eigenvectors

● The smallest eigenvalue of $M$ is greater than the largest eigenvalue of $N$

○ The largest eigenvalue of $M$ is greater than the largest eigenvalue of $N$

We know that $v^T M v \geq \lambda_{\min}(M)$ for all $v$ and $w^T N w \leq \lambda_{\max}(N)$ for all $w$. Thus if $\lambda_{\min}(M) \geq \lambda_{\max}(N)$,

$$v^T M v \geq \lambda_{\min}(M) \geq \lambda_{\max}(N) \geq v^T N v$$
$$v^T (M - N) v \geq 0.$$

These do not require the same eigenvectors. For the last case, consider

$$M = \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix}, \; N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; M - N = \begin{bmatrix} 1 & 0 \\ 0 & -0.5 \end{bmatrix}.$$

$M - N$ is not positive semidefinite since it has an eigenvalue of $-0.5$, but $\lambda_{\max}(M) > \lambda_{\min}(N)$.

**(q)** [3 pts] You are given four sample points $X_1 = [-1, -1]^\top$, $X_2 = [-1, 1]^\top$, $X_3 = [1, -1]^\top$, and $X_4 = [1, 1]^\top$. Each of them is in class C or class D. For what feature representations are the lifted points $\Phi(X_i)$ *guaranteed* to be **linearly separable** (with no point lying exactly on the decision boundary) for every possible class labeling?

○ $\Phi(x) = [x_1, x_2, 1]$

○ $\Phi(x) = [x_1^2, x_2^2, x_1, x_2, 1]$

○ $\Phi(x) = [x_1, x_2, x_1^2 + x_2^2, 1]$

● $\Phi(x) = [x_1^2, x_2^2, x_1 x_2, x_1, x_2, 1]$

Consider the labels $y_1 = 1, y_2 = 0, y_3 = 0, y_4 = 1$. No linear classifier (with or without bias) can classify these samples. Also, $x_1^2 = x_2^2 = 1$ for all $X_i$, and $x_1^2 + x_2^2 = 2$ for all $X_i$.

The last option is a quadratic kernel, which can learn e.g. an elliptical decision boundary in the original sample space to perfectly classify the above labeling.

**(r)** [3 pts] Let $L_i(w)$ be the loss corresponding to a sample point $X_i$ with label $y_i$. The update rule for **stochastic gradient descent** with step size $\epsilon$ is

○ $w_{\text{new}} \leftarrow w - \epsilon \nabla_{X_i} L_i(w)$

● $w_{\text{new}} \leftarrow w - \epsilon \nabla_w L_i(w)$

○ $w_{\text{new}} \leftarrow w - \epsilon \sum_{i=1}^n \nabla_{X_i} L_i(w)$

○ $w_{\text{new}} \leftarrow w - \epsilon \sum_{i=1}^n \nabla_w L_i(w)$

Any option taking the gradient w.r.t. $X_i$ is an incorrect update rule for the weights $w$, so that leaves only the solutions which use $\nabla_w$. Any option which sums over multiple gradients is a batch method, which leaves only one option, the solution.

**(s)** [3 pts] Suppose you have a sample in which each point has $d$ features and comes from class C or class D. The class conditional distributions are $(X_i | y_i = C) \sim N(\mu_C, \sigma_C^2)$ and $(X_i | y_i = D) \sim N(\mu_D, \sigma_D^2)$ for unknown values $\mu_C, \mu_D \in \mathbb{R}^d$ and $\sigma_C^2, \sigma_D^2 \in \mathbb{R}$. The class priors are $\pi_C$ and $\pi_D$. We use 0-1 loss.

● If $\pi_C = \pi_D$ and $\sigma_C = \sigma_D$, then the Bayes decision rule assigns a test point $z$ to the class whose mean is closest to $z$.

● If $\sigma_C = \sigma_D$, then the Bayes decision boundary is always linear.

● If $\pi_C = \pi_D$, then the Bayes decision rule is $r^*(z) = \text{argmin}_{A \in \{C,D\}} \left( |z - \mu_A|^2 / (2\sigma_A^2) + d \ln \sigma_A \right)$

○ If $\sigma_C = \sigma_D$, then QDA will always produce a linear decision boundary when you fit it to your sample.

**(t)** [3 pts] Let $f \in [0, 1]$ be the *unknown*, fixed probability that a person in a certain population owns a dog (how cute!). We model $f$ with a hypothesis $h \in [0, 1]$. Before we observe any data at all, we can't even guess what $f$ might be, so we set our prior probability for $f$ to be the uniform distribution, i.e., $P(f = h) = 1$ over $h \in [0, 1]$. Now, we pick one person from the population, and it turns out that they have a cute little labradoodle named Dr. Frankenstein. Which of the following is true about the posterior probability that $f = h$ given this one sample point?

○ The posterior is uniform over $h \in [0, 1]$.

○ The posterior increases nonlinearly over $h \in [0, 1]$.

● The posterior increases linearly over $h \in [0, 1]$.

○ The posterior is a delta function at 1.

# Q2. [10 pts] The Perceptron Learning Algorithm

The table below is a list of sample points in $\mathbb{R}^2$. Suppose that we run the perceptron algorithm, with a fictitious dimension, on these sample points. We record the total number of times each point participates in a stochastic gradient descent step because it is misclassified, throughout the run of the algorithm.

| $x_1$ | $x_2$ | $y$ | times misclassified |
|---|---|---|---|
| $-3$ | $2$ | $+1$ | $0$ |
| $-1$ | $1$ | $+1$ | $0$ |
| $-1$ | $-1$ | $-1$ | $2$ |
| $2$ | $2$ | $-1$ | $1$ |
| $1$ | $-1$ | $-1$ | $0$ |

**(a)** [5 pts] Suppose that the learning rate is $\epsilon = 1$ and the initial weight vector is $w^{(0)} = (-3, 2, 1)$, where the last component is the bias term. What is the equation of the separating line found by the algorithm, in terms of the features $x_1$ and $x_2$?

At each iteration, the weights are updated by picking a misclassified point and applying the update rule. The learned weights are $w = w^{(0)} + \epsilon \sum_{i=1}^{n} \alpha_i y_i x_i$, where the dual variable $\alpha_i$ is the number of times the $i^{th}$ point is misclassified. Recall that we augment each sample point $x$ with $x_3 = 1$ for the bias. Thus we have $w = (-3, 2, 1) + 2 \cdot -1 \cdot (-1, -1, 1) + 1 \cdot -1 \cdot (2, 2, 1) = (-3, 2, -2)$. Therefore, the equation of the separating line is $\boxed{-3x_1 + 2x_2 - 2 = 0}$

**(b)** [2 pts] In some cases, removing even a single point can change the decision boundary learned by the perceptron algorithm. For which, if any, point(s) in our dataset would the learned decision boundary change if we removed it? Explain your answer.

If we removed either of the two points that were misclassified during training, it would cause a change in the learned decision boundary.

**(c)** [3 pts] How would our result differ if we were to add the additional training point $(2, -2)$ with label $+1$?

The data would no longer be linearly separable, so the perceptron algorithm would not terminate.

# Q3. [10 pts] Quadratic Discriminant Analysis

**(a)** [4 pts] Consider 12 labeled data points sampled from three distinct classes:

Class 0: $\begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} -2 \\ 0 \end{bmatrix}, \begin{bmatrix} 5 \\ 3 \end{bmatrix}, \begin{bmatrix} -3 \\ -5 \end{bmatrix}$ Class 1: $\begin{bmatrix} \sqrt{2} \\ \sqrt{2} \end{bmatrix}, \begin{bmatrix} -\sqrt{2} \\ \sqrt{2} \end{bmatrix}, \begin{bmatrix} 4\sqrt{2} \\ -\sqrt{2} \end{bmatrix}, \begin{bmatrix} -4\sqrt{2} \\ -\sqrt{2} \end{bmatrix}$ Class 2: $\begin{bmatrix} 3 \\ 5 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 8 \\ 6 \end{bmatrix}, \begin{bmatrix} 0 \\ -2 \end{bmatrix}$

For each class $C \in \{0, 1, 2\}$, compute the class sample mean $\mu_C$, the class sample covariance matrix $\Sigma_C$, and the estimate of the prior probability $\pi_C$ that a point belongs to class $C$. (Hint: $\mu_1 = \mu_0$ and $\Sigma_2 = \Sigma_0$.)

Class 0: Mean is $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$, covariance is $\begin{bmatrix} 9.5 & 7.5 \\ 7.5 & 9.5 \end{bmatrix}$, prior is $\frac{1}{3}$

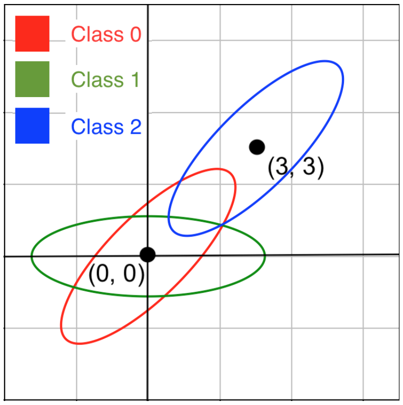
Class 1: Mean is $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$, covariance is $\begin{bmatrix} 17 & 0 \\ 0 & 2 \end{bmatrix}$, prior is $\frac{1}{3}$

Class 2: Mean is $\begin{bmatrix} 3 \\ 3 \end{bmatrix}$, covariance is $\begin{bmatrix} 9.5 & 7.5 \\ 7.5 & 9.5 \end{bmatrix}$, prior is $\frac{1}{3}$

**(b)** [4 pts] Sketch one or more isocontours of the QDA-produced normal distribution or quadratic discriminant function (they each have the same contours) for each class. The isovalues are not important; the important aspects are the centers, axis directions, and relative axis lengths of the isocontours. Clearly label the centers of the isocontours and to which class they correspond.

The ellipses for classes 0 and 1 both need to be centered around the origin. The ellipses for class 0 should be aligned on a 45 degree rotation of the coordinate axes with more variance along the $[1, 1]$ direction than the $[1, -1]$ direction. The ellipses for class 1 should be axis aligned with more variance along the $x$-axis. The ellipses for class 2 must be a translation of the ellipses for class 0.

Note: If incorrect covariance matrices were calculated in the first part, full credit on this part should still be possible so long as each ellipse is centered correctly around the appropriate mean and the variance is in the appropriate directions.



**(c)** [2 pts] Suppose that we apply LDA to classify the data given in part (a). Why will this give a poor decision boundary?

The discriminant functions for classes 0 and 1 would have the exact same mean and covariance, so there would be no decision boundary between them.

# Q4. [10 pts] Ridge Regression with One Feature

We are given a sample in which each point has only one feature. Therefore, our design matrix is a column vector, which we will write $x \in \mathbb{R}^n$ (instead of $X$). Consider the scalar data generation model

$$y_i = \omega x_i + e_i$$

where $x_i \in \mathbb{R}$ is point $i$'s sole input feature, $y_i \in \mathbb{R}$ is its scalar label (a noisy measurement), $e_i \sim \mathcal{N}(0, 1)$ is standard unit-variance zero-mean Gaussian noise, and $\omega \in \mathbb{R}$ is the true, fixed linear relationship that we would like to estimate. The $e_i$'s are independent and identically distributed random variables, and the sole source of randomness. We will treat the design vector $x$ as fixed (not random).

Our goal is to fit a linear model and get an estimate $w_\lambda$ for the true parameter $\omega$. The ridge regression estimate for $\omega$ is

$$w_\lambda = \text{argmin}_{w \in \mathbb{R}} \left( \lambda w^2 + \sum_{i=1}^{n} (y_i - x_i w)^2 \right) \qquad \text{where } \lambda \geq 0.$$

**(a)** [4 pts] Express $w_\lambda$ in terms of $\lambda, S_{xx}$ and $S_{xy}$, where $S_{xx} = \sum_{i=1}^n x_i^2$ and $S_{xy} = \sum_{i=1}^n x_i y_i$.

$$\frac{\partial}{\partial w} \left( \lambda w^2 + \sum_{i=1}^{n} (y_i - x_i w)^2 \right) = 2\lambda w - 2 \sum_{i=1}^{n} y_i x_i + 2 \sum_{i=1}^{n} x_i^2 w$$

Setting the derivative to 0, we have

$$w_\lambda = \frac{S_{xy}}{S_{xx} + \lambda}.$$

**(b)** [5 pts] Compute the squared bias of the ridge estimate $w_\lambda z$ at a test point $z \in \mathbb{R}$, defined to be

$$\text{bias}^2(w_\lambda, z) = (\mathbb{E}[w_\lambda z] - \omega z)^2,$$

where the expectation is taken with respect to the $y_i$'s. Express your result in terms of $\omega, \lambda, S_{xx}$, and $z$. (Hint: simplify the expectation first.)

$$\begin{aligned} \mathbb{E}[w_\lambda] &= \frac{\mathbb{E}[\sum_{i=1}^n x_i y_i]}{S_{xx} + \lambda} \\ &= \frac{\mathbb{E}[\sum_{i=1}^n x_i(\omega x_i + e_i)]}{S_{xx} + \lambda} \\ &= \frac{\omega S_{xx} + \sum_{i=1}^n x_i \cancel{\mathbb{E}[e_i]}}{S_{xx} + \lambda} \\ &= \frac{\omega S_{xx}}{S_{xx} + \lambda}. \end{aligned}$$

Therefore,

$$(\mathbb{E}[w_\lambda z] - \omega z)^2 = (\mathbb{E}[w_\lambda] - \omega)^2 z^2 = \left( -\frac{\omega\lambda}{S_{xx} + \lambda} \right)^2 z^2 = \frac{\omega^2 \lambda^2}{(S_{xx} + \lambda)^2} z^2.$$

**(c)** [1 pt] What will the bias be if we are using ordinary least squares, i.e., $\lambda = 0$?

The bias is zero if $\lambda = 0$.

# Q5. [10 pts] Logistic Regression with One Feature

We are given another sample in which each point has only one feature. Consider a binary classification problem in which sample values $x \in \mathbb{R}$ are drawn randomly from two different class distributions. The first class, with label $y = 0$, has its mean to the left of the mean of the second class, with label $y = 1$. We will use a modified version of logistic regression to classify these data points. We model the posterior probability at a test point $z \in \mathbb{R}$ as

$$P(y = 1|z) = s(z - \alpha),$$

where $\alpha \in \mathbb{R}$ is the sole parameter we are trying to learn and $s(\gamma) = 1/(1 + e^{-\gamma})$ is the logistic function. The decision boundary is $z = \alpha$ (because $s(z) = \frac{1}{2}$ there).

We will learn the parameter $\alpha$ by performing gradient descent on the logistic loss function (a.k.a. cross-entropy). That is, for a data point $x$ with label $y \in \{0, 1\}$, we find the $\alpha$ that minimizes

$$J(\alpha) = -y \ln s(x - \alpha) - (1 - y) \ln(1 - s(x - \alpha)).$$

**(a)** [5 pts] Derive the stochastic gradient descent update for $J$ with step size $\epsilon > 0$, given a sample value $x$ and a label $y$. Hint: feel free to use $s$ as an abbreviation for $s(x - \alpha)$.

By the chain rule,

$$\frac{d}{d\alpha} s(x - \alpha) = -s(1 - s).$$

Hence,

$$\begin{aligned} J'(\alpha) &= \frac{ys(1 - s)}{s} - \frac{(1 - y)s(1 - s)}{1 - s} \\ &= y(1 - s) - (1 - y)s \\ &= y - s. \end{aligned}$$

So the stochastic gradient descent update rule is

$$\alpha^{(t+1)} \leftarrow \alpha^{(t)} + \epsilon(s(x - \alpha) - y).$$

**(b)** [3 pts] Is $J(\alpha)$ convex over $\alpha \in \mathbb{R}$? Justify your answer.

Continuing from the last part,

$$J''(\alpha) = \frac{d}{d\alpha}(y - s) = s(1 - s).$$

As the logistic function is always in the range $(0, 1)$, $s(1 - s)$ is always positive, so $J(\alpha)$ is convex. (Moreover, it's strictly convex and hence admits at most one solution.)

**(c)** [2 pts] Now we consider multiple sample points. As $d = 1$, we are given an $n \times 1$ design matrix $X$ and a vector $y \in \mathbb{R}^n$ of labels. Consider batch gradient descent on the cost function $\sum_{i=1}^{n} J(\alpha; X_i, y_i)$. There are circumstances in which this cost function does not have a minimum over $\alpha \in \mathbb{R}$ at all. What is an example of such a circumstance?

If all the sample points are of only one class, then $J(\alpha)$ is either monotonically increasing or monotonically decreasing over $\alpha \in \mathbb{R}$.