CS 189/289A Introduction to Machine Learning
Spring 2025 Jonathan Shewchuk Midterm

- Please do not open the exam before you are instructed to do so. Fill out the blanks below now.
- **Electronic devices are forbidden on your person**, including phones, laptops, tablet computers, headphones, and calculators. Turn your cell phone off and **leave all electronics at the front of the room**, or **risk getting a zero** on the exam. Exceptions are made for car keys and devices needed because of disabilities.
- When you start, the **first thing you should do** is **check that you have all 10 pages and all 5 questions**. The second thing is to please **write your initials at the top right of every page after this one** (e.g., write "JS" if you are Jonathan Shewchuk).
- The exam is closed book, closed notes except your one cheat sheet.
- You have **110 minutes**. (If you are in the Disabled Students' Program and have an allowance of 150% or 200% time, that comes to 165 minutes or 220 minutes, respectively.)
- Mark your answers on the exam itself in the space provided. Do **not** attach any extra sheets. If you run out of space for an answer, write a note that your answer is continued on the bottom of page 4 or 10.
- The total number of points is 100. There are 12 multiple choice questions worth 4 points each, and 4 written questions worth a total of 52 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	
Name and SID of student to your left	
Name and SID of student to your right	

Q1. [48 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] Select the statements true of **every symmetric, positive semidefinite matrix** A.

- ● A: For all $x \in \mathbb{R}^n$, $x^\top Ax \geq 0$.
- ○ B: All the eigenvalues of A are strictly positive.
- ● C: There exists a matrix $U \in \mathbb{R}^{n \times n}$ such that $A = UU^\top$.
- ○ D: A is invertible.

(A) **True**. See Homework 2 Question 3.2.

(B) **False**. PSD matrices have non-negative eigenvalues, not strictly positive.

(C) **True**. See Homework 3 Problem 3.2.

(D) **False**. PSD matrices are only invertible if they are also positive definite.

(b) [4 pts] The **hard-margin support vector machine** (SVM) optimization problem can be interpreted as ...

- ● A: maximizing the distance from the closest training point to the decision boundary.
- ○ B: maximizing the sample variance of the training points after they are projected onto the decision boundary.
- ● C: minimizing the length of the weight vector (not including the bias term), subject to constraints.
- ○ D: minimizing the mean-squared distance from the training points to the decision boundary.

A: True. The hard-margin SVM objective is to maximize the margin, which is defined as the distance between the decision boundary and the closest points on either side (i.e., the support vectors).

B: False, but PCA fits this description.

C: True. The SVM objective is to minimize $\|w\|^2$.

D: False, but both PCA and least-squares linear regression fit this description.

(c) [4 pts] Given a training set for a two-class classification problem, we use a **soft-margin support vector machine** (SVM) augmented by lifting each training point X_i to a lifted point $\Phi(X_i)$ with all the monomials of degree zero through three, so that the decision function can be any cubic function of x. Select the true statements.

- ● A: If the training set is small, the lifted training points are more vulnerable to overfitting than the original, unmodified training points.
- ● B: It is possible for the decision boundary in the original space to be linear.
- ○ C: The data in the lifted feature space are guaranteed to be linearly separable.
- ● D: It is possible for the decision boundary in the original space to be a hypersphere.

A: True. Cubic transformations increase the feature space's dimension a lot, which may lead to overfitting, particularly if the dataset is small or noisy.

B: True. It is possible that all the weights for the quadratic and cubic terms will be zero after optimization.

C: False. There are lots of (large) data sets that can't be separated by cubic decision functions.

D: True. For example, if the decision function is $\|x\|^2 - 1$.

(d) [4 pts] Consider a **soft-margin support vector machine** (SVM) modified to minimize the objective function

$$\|w\|^2 + C_+ \sum_{y_i=1} \xi_i + C_- \sum_{y_i=-1} \xi_i,$$

where $C_+ > 0$ is the penalty for the slack of each positive training point and $C_- > 0$ is a different penalty for the slack of each negative training point. The first summation sums the slack over all the positive training points, and the second sums the slack over all the negative training points. The constraints are the same as in a standard soft-margin SVM. Assume there is at least one training point from each class. Select the true statements.

- A: If some positive training points have nonzero slack, then as C_+ increases (with C_- unchanging), the decision boundary *tends* to shift toward the negative training points.
- B: Validation is an effective way to choose C_+ and C_-.
- C: If $C_+ = 2$ and $C_- = 1$, the decision boundary is the same as that produced by a standard SVM with $C = 1$ trained with every positive training point duplicated (so there are two of each) and the negative training points unchanged (just one of each).
- D: The modified objective function is convex.

A: True. A higher penalty for positive training points tends to push the boundary away from them, toward negative training points.

B: Of course. How else would you choose them?

C: True. Both machines have the same objective function. Duplicated points will have the same slack as they have in the modified SVM.

D: True; with respect to the variables w, α, and $\xi_i, i \in [1, n]$, the cost function has a positive semidefinite Hessian everywhere. Specifically, it's a diagonal matrix with a 2 in every diagonal entry corresponding to a component of w and a 0 everywhere else.

(e) [4 pts] We run **gradient descent** on the quadratic function $f(w) = w^\top H w$ from a random starting point with a step size of $\epsilon = 0.5$. What value of H is most likely to lead to the fastest convergence?

○ A: $H = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$.

● B: $H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

○ C: $H = \begin{bmatrix} 10 & 0 \\ 0 & 0.1 \end{bmatrix}$.

○ D: $H = \begin{bmatrix} 0.1 & 0.1 \\ 0.1 & 0.1 \end{bmatrix}$.

With option B, $\nabla f = (2w_1, 2w_2)$ and gradient descent takes only one step. Options A and C cause gradient descent to diverge because the step size is too large (their optimal step size is $\epsilon = 0.05$). Option D will converge, but slowly.

(f) [4 pts] Consider the **least-squares linear regression** problem of finding the weight vector $w \in \mathbb{R}^d$ that minimizes the cost function $\|Xw - y\|^2$, where X is the $n \times d$ design matrix and $y \in \mathbb{R}^n$ is a vector of labels. Select the true statements.

● A: There exists at least one minimizer $\hat{w}$.

● B: Let $\hat{w}$ be a minimizer. Then the set of all minimizers is $\{\hat{w} + \Delta : \Delta \in \text{nullspace } X\}$.

● C: There is no more than one minimizer in the rowspace of X.

● D: If $n < d$, there are infinitely many minimizers.

A: True. This follows from the fact that the normal equations $X^\top X w = X^\top y$ provide necessary and sufficient conditions for the minimizer. $\text{col } X^\top X = \text{col } X^\top$, so there always exists a solution. This fact and the following two were derived in discussion 6.

B: True. (See discussion 6.)

C: True. (See discussion 6.)

D: True: if $n < d$, then null X is not just the trivial subspace $\{0\}$; it has infinitely many points. Note also that $X^\top X$ cannot have full rank if $n < d$.

(g) [4 pts] Select the true statements about **linear regression**.

● A: The cost function for ridge regression is minimized by just one step of Newton's Method.

○ B: Added quadratic features will always cause the model to obtain a lower cost on the validation set.

○ C: The cost function for Lasso is minimized by just one step of Newton's Method.

○ D: The cost function for Lasso includes a regularization term $\lambda \|w\|_2^2$.

A is True as the cost of RSS as well as an L2 penalty term are quadratic, and Newton's method finds an optimal solution if the cost is quadratic.

B is False. Adding new features increases the variance, so it's possible that the validation error will drop.

C is False, as Lasso's cost function is not quadratic (nor even smooth).

D is False. The Lasso regularization term is $\lambda \|w'\|_1$..

(h) [4 pts] Select the true statements about adding and removing **features**.

○ A: Adding features *tends* to increase bias.

○ B: Removing features with no predictive power *tends* to improve training accuracy.

● C: Adding features *tends* to increase variance.

● D: Removing features with no predictive power *tends* to improve validation accuracy.

A: False. Adding features usually reduces bias, as it increases the flexibility of the model.

B: False. Removing irrelevant features generally does not improve training accuracy, though it often improves validation and test accuracy.

C: True. Adding features tends to increase variance because it adds another degree of freedom in which the data can vary.

D: True. Removing irrelevant features reduces the variance (while not reducing bias much if at all), thus reducing the tendency of the model to overfit to the training data. This then improves the validation accuracy.

(i) [4 pts] Select the true statements about **binary classification** when each of the two classes is (precisely) **normally distributed**. Assume **no added features** are used other than the input features, and the loss is always zero for correct predictions. For LDA and QDA, assume there is at least one training point from each class.

● A: The LDA decision boundary is always linear, even with an asymmetric loss function.

○ B: The Bayes optimal decision boundary does not change if we replace an asymmetric loss function $L(\hat{y}, y)$ with the loss $L(y, \hat{y})$.

○ C: The QDA decision boundary is always nonlinear, even with a symmetric loss function.

● D: The Bayes optimal decision boundary does not change if we replace the loss function $L(\hat{y}, y)$ with the loss $2L(\hat{y}, y)$.

A: True, the LDA decision boundary is always linear. An asymmetric loss function only translates the decision boundaries.

B: False; reversing an asymmetric loss adds a constant to the decision function.

C: False, the QDA decision boundary can be linear if the sample covariances for all the classes are the same.

D: True:

$$\begin{aligned} r^*(x) &= \arg\min_{r(x)} \sum_{i=1}^{k} P(Y=i) \int 2L(r(x), y=i) f_{X|Y=i}(x) dx \\ &= \arg\min_{r(x)} 2 \sum_{i=1}^{k} P(Y=i) \int L(r(x), y=i) f_{X|Y=i}(x) dx \\ &= \arg\min_{r(x)} \sum_{i=1}^{k} P(Y=i) \int L(r(x), y=i) f_{X|Y=i}(x) dx. \end{aligned}$$

(j) [4 pts] Select the true statements about **Gaussian discriminant analysis** (GDA).

○ A: Like logistic regression, GDA directly fits a function modeling the posterior probability to the training points.

○ B: The **decision function** that LDA uses for binary classification has $\Theta(d^2)$ parameters (i.e., the number of decision function parameters is proportional to d^2), where d is the dimension of the feature space.

○ C: One reason to use QDA is because LDA fails to produce any classifier at all if the training points are not linearly separable.

● D: GDA does not require us to use an iterative optimization method because the maximum likelihood estimates have closed-form solutions.

A: False. GDA is a generative model. It learns $P(X = x|Y = C)$ and $P(X = x|Y = D)$ (by fitting functions), but then it calculates the posterior probability from those, not by a direct fit to data.

B: False; that's QDA.

C: False. Of the algorithms we've studied, only the perceptron algorithm and hard-margin SVMs require linearly separable points to produce a classifier.

D: True.

(k) [4 pts] Consider the problem of finding a vector $w \in \mathbb{R}^2$ that minimizes

$$f(w) = w^\top Q w - b^\top w, \quad \text{where} \quad Q = \begin{bmatrix} -10 & -1 \\ -1 & -10 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 6 \\ 2 \end{bmatrix}.$$

Select the true statements about iterative algorithms on f from a starting point of $w^{(0)} = [0 \quad 0]^\top$.

- ● A: Newton's method will converge to a critical point of f.
- ○ B: Newton's method will converge to a local minimum of f.
- ○ C: Gradient descent will converge to a local minimum of f.
- ○ D: We can find a local minimum of f by calculus and solving a system of two linear equations.

A: As f is quadratic, Newton's method jumps to the critical point of f in one iteration.

B, C, and D must all be wrong because f does not have a minimum!

(l) [4 pts] A factory produces widgets, but 10% of the widgets are defective.

The factory uses an automated scanner that predicts class "defective" or class "not defective" for each widget. If a widget is defective, the scanner correctly predicts "defective" 90% of the time (true positive rate). If a widget is not defective, the scanner incorrectly predicts "defective" 5% of the time (false positive rate). Discarding a good widget causes a loss of 2 dollars. Allowing a defective widget to pass on to customers causes a loss of 10 dollars.

A newly scanned widget is predicted to be defective. Select the true statements.

- ● A: To compute the probability that this widget is actually defective, we can use Bayes' Theorem.
- ○ B: The probability that this widget is actually defective is less than 50%.
- ● C: To minimize the overall expected loss, the factory should discard this widget.
- ● D: If the factory discards all widgets with prediction "defective," the expected loss is more than 0.5 dollars for each widget with prediction "defective."

Applying **Bayes' theorem**, we compute the probability that a widget is actually defective given that it was flagged by the scanner.

- $P(D) = 0.1$ be the prior probability of a widget being defective.
- $P(\neg D) = 0.9$ be the prior probability of a widget being non-defective.
- $P(F \mid D) = 0.9$ be the probability that a defective widget is flagged.
- $P(F \mid \neg D) = 0.05$ be the probability that a non-defective widget is flagged.

Using the **law of total probability** to compute $P(F)$:

$$P(F) = P(F \mid D)P(D) + P(F \mid \neg D)P(\neg D)$$

$$P(F) = (0.9 \times 0.1) + (0.05 \times 0.9) = 0.09 + 0.045 = 0.135$$

Now, applying Bayes' theorem:

$$P(D \mid F) = \frac{P(F \mid D)P(D)}{P(F)}$$

$$P(D \mid F) = \frac{0.9 \times 0.1}{0.135} = \frac{0.09}{0.135} = 0.6667$$

Thus, the probability that a flagged widget is actually defective is 66.67%.

B is false: The probability is **greater** than 50%, not less.

A is true: Bayes' theorem was essential in computing this probability.

Expected losses for each decision:

- **Discard flagged widget**: expected loss = $P(\neg D \mid F) \times 2 = (1 - 0.6667) \times 2 = 0.3333 \times 2 = 0.67$.
- **Keep flagged widget**: expected loss = $P(D \mid F) \times 10 = 0.6667 \times 10 = 6.67$.

C is true: Since $6.67 > 0.67$, it is optimal to discard flagged widgets in this case.

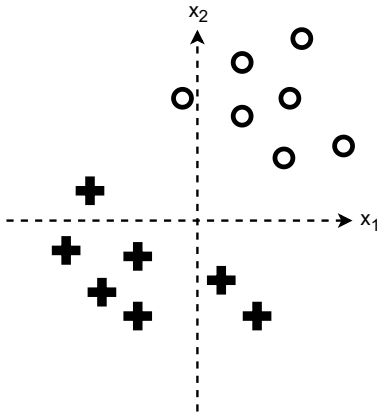
D is true: 67 cents is more than 50 cents.

Extra space: if you need extra space for your answer to a written problem, you may write here. **Be sure to write "see page 4" under the unfinished answer!** (If needed, there is more extra space on page 10.)

Q2. [10 pts] Regularized Logistic Regression

We apply **logistic regression with regularization** to a binary (two-class) classification problem in a two-dimensional feature space ($d = 2$). We are using a fictitious dimension to support a bias term α; hence, the weight vector is $w = [w_1 \quad w_2 \quad \alpha]^\top$, and each training point has a "1" appended to it. What happens if we apply ℓ_2 regularization to only a single weight?

Our training points appear below. The crosses have labels of $y_i = +1$ and the circles have labels of $y_i = 0$.



(a) [6 pts] Suppose we regularize w_2. We (try to) minimize the cost function

$$J(w) = \left(-\sum_{i=1}^{n} \left(y_i \ln s(X_i \cdot w) + (1 - y_i) \ln \left(1 - s(X_i \cdot w)\right)\right)\right) + \lambda w_2^2,$$

where $s(\gamma)$ is the logistic function and $\lambda > 0$ is the regularization hyperparameter.

As λ increases from zero to infinity, **explain how the decision boundary changes**, **explain how the number of misclassified training points changes**, and **explain why**.

At $\lambda = 0$ there are no misclassified points, as logistic regression gives the maximum margin classifier on linearly separable points. As λ increases, w_2 is forced toward zero, so the decision boundary rotates clockwise, approaching a vertical line in the limit. The number of misclassified points increases to two or three. (It's not obvious whether it's two or three, but for the purpose of grading, we don't care.)

(b) [4 pts] Suppose we regularize α. We (try to) minimize the cost function

$$J(w) = \left(-\sum_{i=1}^{n} \left(y_i \ln s(X_i \cdot w) + (1 - y_i) \ln \left(1 - s(X_i \cdot w)\right)\right)\right) + \lambda \alpha^2.$$

As λ increases from zero to infinity, **explain how the decision boundary changes**, **explain how the number of misclassified points changes**, and **explain why**.

Again, at $\lambda = 0$ we have the maximum margin classifier and no misclassified points. As λ increases, α is forced toward zero, so the decision boundary shifts toward the origin, approaching the origin in the limit. The number of misclassified points does not change: it remains zero, because a line through the origin can separate the classes. (It's not obvious whether the decision boundary rotates a little bit, but for the purpose of grading, we don't care.)

Q3. [14 pts] A Bayes Decision Rule for Biased Coins

I have two biased coins. Coin 1 comes up heads with probability $p_1 = 2/3$, and Coin 2 comes up heads with probability $p_2 = 3/4$. I hand you a coin. You don't know which coin I handed you, but you know that there's a 2/5 chance that I handed you Coin 1, and a 3/5 chance that I handed you Coin 2. You flip the coin n times, then you try to predict which coin it is.

(a) [6 pts] You flip the coin I handed you n times and obtain x heads. **Write a Bayes decision rule** $r^*(x)$ whose value is either 1 or 2 for any $x \in [0, n]$, predicting which coin I handed you. (Hint: use an inequality to decide which value to return.) We are using a 0-1 loss function (i.e., we want to maximize the probability of a correct prediction).

Write your answer **in terms of the probability mass functions (PMFs) for the binomial distribution—$P(X = x|Y = 1)$ for Coin 1 and $P(X = x|Y = 2)$ for Coin 2—and the appropriate prior probabilities**. Do not substitute the binomial distribution's PMF formula yet; you'll do that in part (b). Otherwise, **simplify as much as possible**.

The Bayes decision rule selects the class that minimizes the risk (i.e., the expected loss). If we assume the loss is symmetric, then Bayes' rule simply selects the class with the larger posterior probability. For this example, our posterior probabilities are

$$P(Y = i|X = x) = \frac{P(X = x|Y = i)P(Y = i)}{P(X = x)}, \; i \in \{1, 2\}.$$

The Bayes decision rule says to choose the class with the biggest posterior probability. You choose coin 1 if

$$\frac{P(X = x|Y = 1)P(Y = 1)}{P(X = x)} > \frac{P(X = x|Y = 2)P(Y = 2)}{P(X = x)}.$$

So the Bayes decision rule is

$$r^*(x) = \begin{cases} 1 & \text{if } P(X = x|Y = 1)P(Y = 1) > P(X = x|Y = 2)P(Y = 2), \\ 2 & \text{otherwise.} \end{cases}$$

It's also fine to write the inequality as $2P(X = x|Y = 1) > 3P(X = x|Y = 2)$.

(b) [4 pts] Recall the probability mass function (PMF) for the binomial distribution: the probability of obtaining x heads in n flips with coin i is

$$P(X = x|Y = i) = \binom{n}{x} p_i^x (1 - p_i)^{n-x}.$$

If $n = 2$, **write the values of $r^*(0)$, $r^*(1)$, and $r^*(2)$. Show your work.**

The inequality becomes

$$\frac{2}{5}\binom{n}{x}\left(\frac{2}{3}\right)^x\left(\frac{1}{3}\right)^{n-x} > \frac{3}{5}\binom{n}{x}\left(\frac{3}{4}\right)^x\left(\frac{1}{4}\right)^{n-x}.$$

Setting $n = 2$, this simplifies to $2^{x+5} > 3^{x+3}$. Hence $r^*(0) = 1$ and $r^*(1) = r^*(2) = 2$.

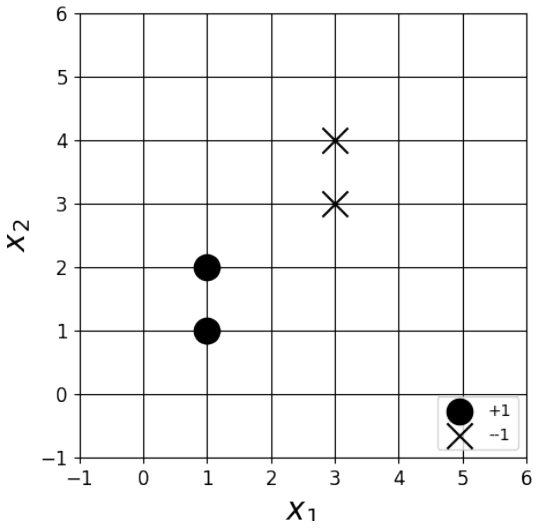
(c) [4 pts] Consider the decision rule $r(x) = 1$, meaning a decision rule that always guesses "Coin 1" regardless of the number of heads x. **Write an expression for the risk** $R(r)$—the risk of the decision rule $r(x)$, assuming you use the 0-1 loss function. **Explain your answer.**

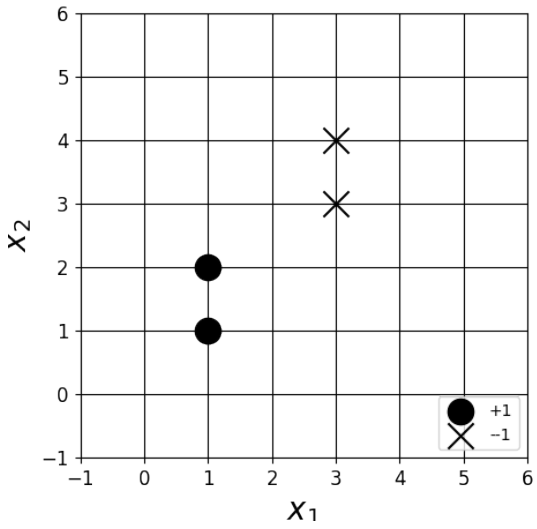
$R(r) = 3/5$, because that's the probability I handed you Coin 2.

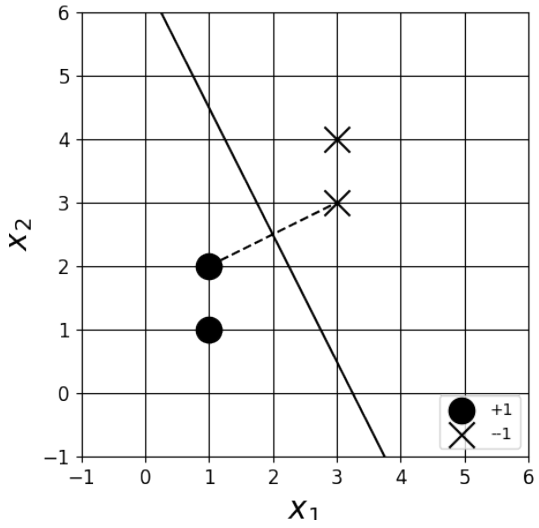
Q4. [14 pts] Sensitivity of Maximum Margin Classifiers

(a) [4 pts] Below is an illustration of a two-dimensional feature space and four training points, two of class O (label $y_i = +1$) and two of class X (label $y_i = -1$). **Draw the decision boundary of the maximum margin classifier** (also known as a hard-margin support vector machine).

(Note: there are two copies below so if you mess it up, you can start over. But you only need to draw on one. If you draw on both, tell us which one to grade. If you draw on both and don't tell us, we'll grade the right one.)







(b) [4 pts] **Compute the maximum margin classifier's optimal parameters w_1, w_2, and α.**

The points $(1, 2)$ and $(3, 3)$ are the support vectors for the $+1$ and -1 class, respectively. We can draw the distance vector between the two support vectors. We know that the w is some multiple of the distance vector, and that the decision boundary is the perpendicular bisector of the distance vector. This implies $w = \lambda \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and based on the y-intercept, $\alpha = -6.5\lambda$. Using the

fact that $w^T \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \alpha = 1$, this implies $\lambda = -0.4$, and

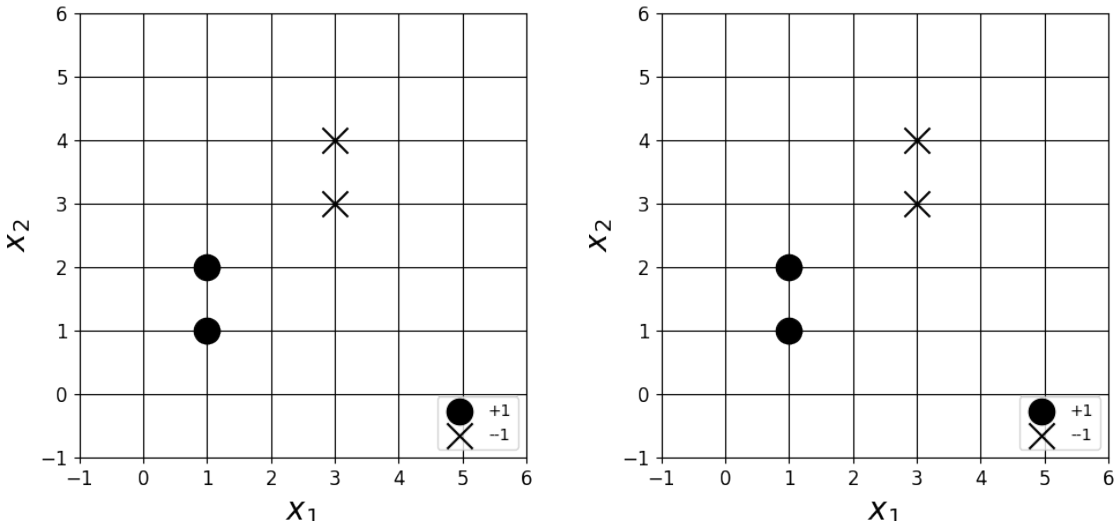
$$w = \begin{bmatrix} -0.8 \\ -0.4 \end{bmatrix} \quad \alpha = 2.6$$

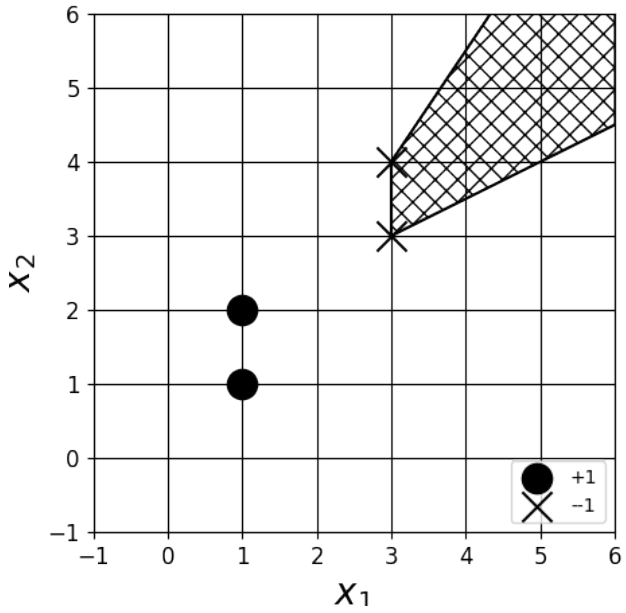
.

Alternatively, we can solve the systems of equations using the fact that the support vectors lie on the margins and the midpoint of the support vectors lies on the boundary,

$$w^T \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \alpha = 1$$
$$w^T \begin{bmatrix} 3 \\ 3 \end{bmatrix} + \alpha = -1$$
$$w^T \begin{bmatrix} 2 \\ 2.5 \end{bmatrix} + \alpha = 0.$$

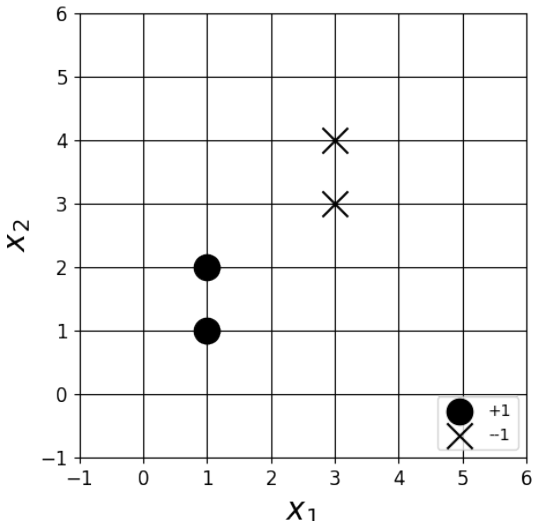
(c) [3 pts] Consider adding one additional training point X_5 of class O (label $y_5 = +1$). Draw as a **shaded polygon** the **set of all points in the illustration** such that there will be **no** maximum margin classifier (no solution to the constrained optimization problem) if you place X_5 there.

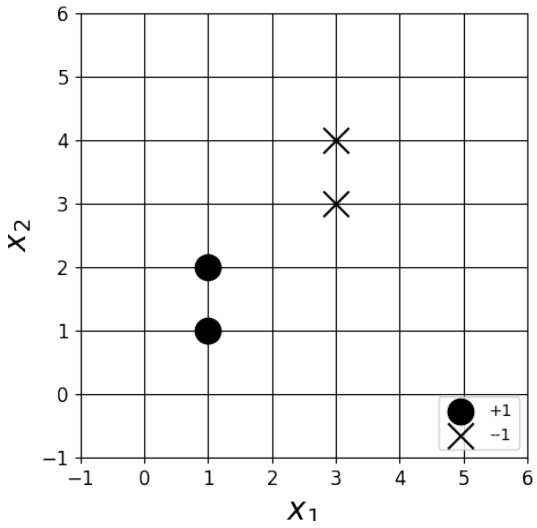


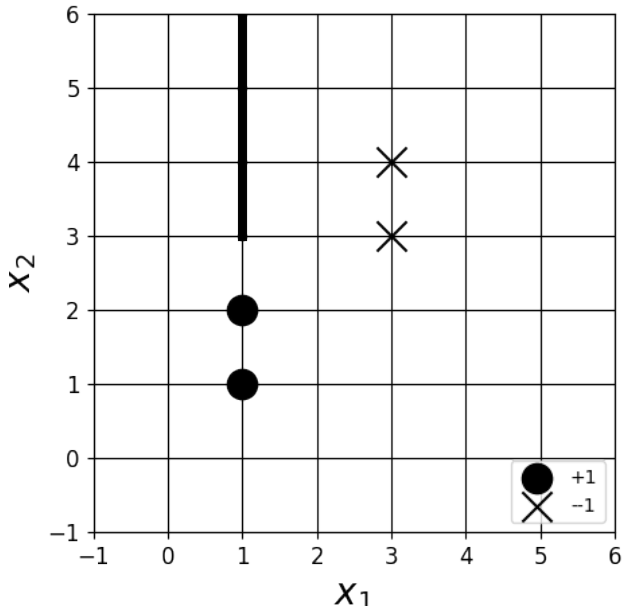


Any point of class O placed in this polygon will render the training points no longer linearly separable.

(d) [3 pts] Consider adding one additional training point X_5 of class O (label $y_5 = +1$). Draw the **set of all points in the illustration** such that if you place X_5 there, **all five training points** will be support vectors of the **updated** maximum margin classifier.







Any point of class O placed on the ray $x_1 = 1$, $x_2 \geq 3$ will cause the updated maximum margin classifier to have the decision boundary $x_1 = 2$, with all five training points at a distance of exactly 1 from the decision boundary.

Q5. [14 pts] Axis-Aligned Quadratic Discriminant Analysis

Consider a version of **quadratic discriminant analysis** (QDA) in which we constrain each class's estimated covariance matrix to be **diagonal**. Let d be the number of features of each training point. For each class C, we use the training points of class C to estimate a mean and a $d \times d$ covariance matrix

$$\hat{\mu}_{\mathrm{C}} = \begin{bmatrix} \hat{\mu}_{\mathrm{C},1} \\ \hat{\mu}_{\mathrm{C},2} \\ \vdots \\ \hat{\mu}_{\mathrm{C},d} \end{bmatrix}, \qquad \hat{\Sigma}_{\mathrm{C}} = \begin{bmatrix} \hat{\sigma}^2_{\mathrm{C},1} & 0 & \dots & 0 \\ 0 & \hat{\sigma}^2_{\mathrm{C},2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{\sigma}^2_{\mathrm{C},d} \end{bmatrix}. \tag{1}$$

(a) [5 pts] Write the probability density function (PDF) $f(x)$ for the multivariate normal distribution $\mathcal{N}(\hat{\mu}_{\mathrm{C}}, \hat{\Sigma}_{\mathrm{C}})$ with the parameters specified by (1), then **show how to factor** it into d univariate normal distribution PDFs. **Show your work.**

$$\begin{aligned} f(x) &= \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x-\hat{\mu}_{\mathrm{C}})^\top \Sigma^{-1} (x-\hat{\mu}_{\mathrm{C}})\right) \\ &= \frac{1}{\sqrt{(2\pi)^d \prod_{i=1}^d \sigma^2_{\mathrm{C},i}}} \exp\left(-\frac{1}{2}\sum_{i=1}^d \sigma^{-2}_{\mathrm{C},i}(x_i-\hat{\mu}_{\mathrm{C},i})^2\right) \\ &= \prod_{i=1}^d \frac{1}{\sqrt{2\pi}\sigma_{\mathrm{C},i}} \exp\left(-\frac{1}{2\sigma^{-2}_{\mathrm{C},i}}(x_i-\hat{\mu}_{\mathrm{C},i})^2\right) = \prod_{i=1}^d f_i(x_i). \end{aligned}$$

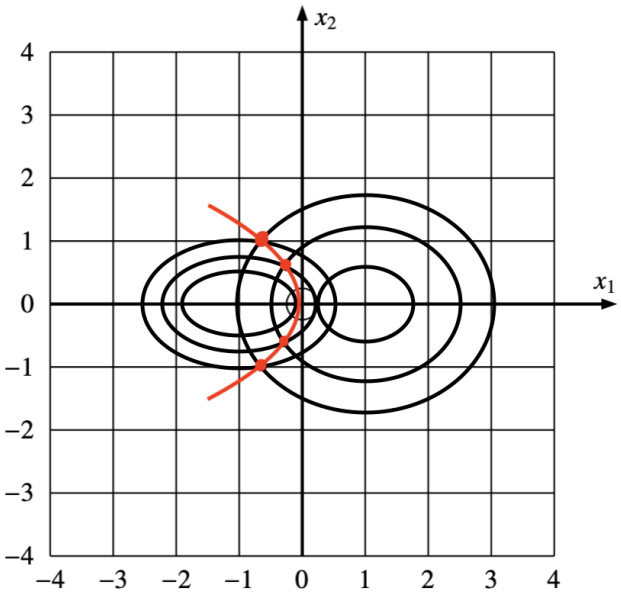
Where we get the second equality from the fact that the determinant of a diagonal matrix is the product of their diagonal entries.

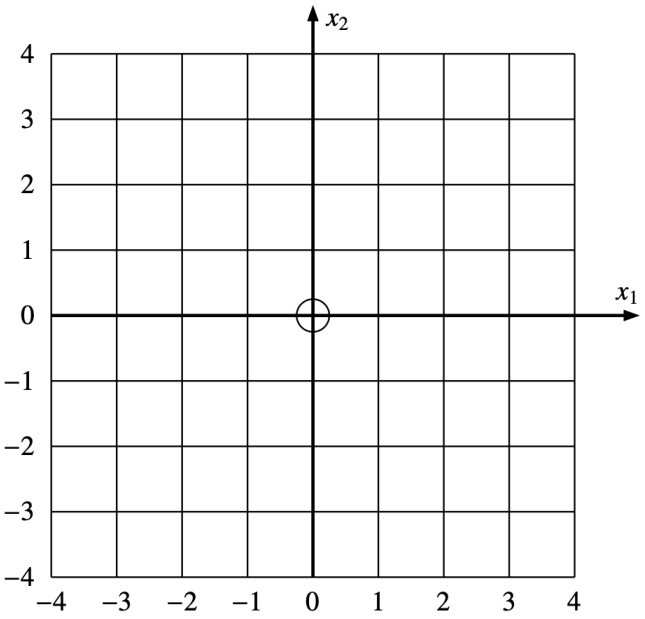
Here, $f_i(x_i)$ refers to the univariate pdf with mean $\hat{\mu}_{\mathrm{C},i}$ and standard deviation $\sigma_{\mathrm{C},i}$.

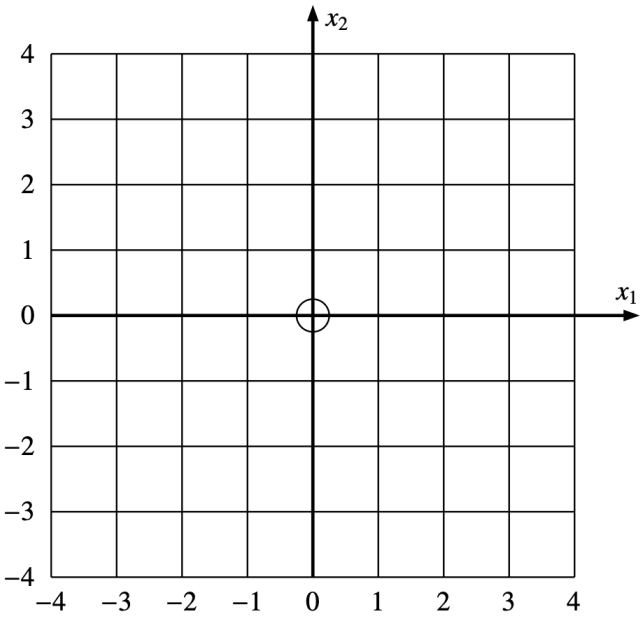
(b) [5 pts] Draw some isocontours for the PDFs of two plausible normal distributions $\mathcal{N}(\mu_{\mathrm{C}}, \Sigma_{\mathrm{C}})$ and $\mathcal{N}(\mu_{\mathrm{D}}, \Sigma_{\mathrm{D}})$ with **diagonal convariance matrices**, such that the Bayes optimal decision boundary between class C and class D is **curved, not linear**. To confirm that the decision boundary is not linear, you must (1) draw several reasonably-spaced isocontours for each class, (2) identify at least three points of the decision boundary where an isocontour of class C's PDF intersects a isocontour of class D's PDF with the same isovalue, such that the points clearly don't lie on a line. You don't need to specify any numbers, but make your isocontour spacing look regular (so the decision boundary points look plausible).

(Note: there are two copies below so if you mess it up, you can start over. But you only need to draw on one. If you draw on both, tell us which one to grade. If you draw on both and don't tell us, we'll grade the right one.)

Any axis-aligned elliptical/spherical isocontours would be valid as long as they're different from one another (as that would create a linear decision boundary). Points are awarded for identifying 3 points–the example graph provides 4 along with the decision boundary for clarity.







(c) [4 pts] To perform quadratic disciminant analysis, we need estimates $\hat{\mu}_C$ and $\hat{\Sigma}_C$ of the parameters of class C's distribution. You are given a set $X_1, X_2, \ldots, X_n$ of training points, all of class C. We can use maximum likelihood estimation (MLE) to choose estimates. As usual, MLE tells us to estimate the mean of class C's distribution by taking the sample mean of the points, $\hat{\mu}_C = \frac{1}{n}\sum_{i=1}^{n} X_i$. **What formula should we use to compute the parameters $\hat{\sigma}^2_{C,i}$?**

There are two ways you could answer this question. The hard way is to apply MLE directly to (1) and grind through the math. The easy way is to use your answer to part (a) to show that the likelihood function is related to a problem we already solved in class (applied to the univariate normal distribution), and we can use our solution from class here. **Explain that relationship, and explain how we can maximize the likelihood function we face now by reusing the maximizer we learned in class for a simpler likelihood function.** Your explanation can mix English and math however you like; the only math that is required is your formula for $\hat{\sigma}^2_{C,i}$.

(*Hint: The likelihood is a product of products. Think about how to regroup the factors by their parameters.*)

METHOD 1

The likelihood function is a product of PDFs, $\mathcal{L}(\hat{\sigma}_{C,1}, \ldots, \hat{\sigma}_{C,d}) = f(X_1)f(X_2)\ldots f(X_n)$. From part (a), each of those PDFs is a product of PDFs; for instance, $f(X_1) = f_1(X_{11})f_2(X_{12})\cdots f_d(X_{1d})$. Observe that the parameter $\hat{\sigma}_{C,i}$ appears only in the PDF f_i, which has no other parameter. We can regroup those factors to write $\mathcal{L} = (f_1(X_{11})f_1(X_{21})\cdots f_1(X_{n1}))\ldots(f_d(X_{1d})f_d(X_{2d})\cdots f_d(X_{nd}))$. This is a product of independent MLE problems, each with a different parameter. Hence we can maximize each $f_1(X_{i1})f_1(X_{i1})\cdots f_1(X_{i1}))$ separately. We derived the answer in class: the optimal parameter is

$$\hat{\sigma}^2_{C,i} = \frac{1}{n}\sum_{j=1}^{n} \|X_{ji} - \hat{\mu}_{C,i}\|^2,$$

which is the sample variance for each feature of X.

METHOD 2 [DIRECT]

We can alternatively also solve it directly by applying MLE on our log likelihood function.

We start by writing out our full likelihood function as

$$L(X_C \mid \mu_C, \Sigma_C) = \prod_{j=1}^{N_C} f(x_j)$$

and take the log likelihood to make taking the derivative easier,

$$\ln(L(X_C \mid \mu_C, \Sigma_C)) = \sum_{j=1}^{N_C} \ln f(x_j)$$

$$\ln(L(X_C \mid \mu_C, \Sigma_C)) = \sum_{j=1}^{N_C}\left(-\frac{d}{2}\ln(2\pi) - \sum_{i=1}^{d}\ln\sigma_{C,i} - \frac{1}{2}\sum_{i=1}^{d}\frac{(x_{j,i}-\mu_{C,i})^2}{\sigma^2_{C,i}}\right).$$

To find the maximum likelihood estimate for $\sigma^2_{C,i}$, we differentiate the log-likelihood function with respect to each $\sigma^2_{C,i}$ and solve for the optimal value.

$$\frac{\partial}{\partial\sigma^2_{C,i}}\ln(L(X_C \mid \mu_C, \Sigma_C)) = -\frac{N_C}{2\sigma^2_{C,i}} + \frac{1}{2}\sum_{j=1}^{N_C}\frac{(x_{j,i}-\mu_{C,i})^2}{\sigma^4_{C,i}}.$$

Setting this equal to zero:

$$-\frac{N_C}{2\sigma^2_{C,i}} + \frac{1}{2}\sum_{j=1}^{N_C}\frac{(x_{j,i}-\mu_{C,i})^2}{\sigma^4_{C,i}} = 0.$$

Rearranging:

$$\sigma_{C,i}^2 = \frac{1}{N_C} \sum_{j=1}^{N_C} (x_{j,i} - \mu_{C,i})^2.$$

Thus, the maximum likelihood estimate (MLE) for $\sigma_{C,i}^2$ is:

$$\hat{\sigma}_{C,i}^2 = \frac{1}{N_C} \sum_{j=1}^{N_C} (x_{j,i} - \hat{\mu}_{C,i})^2,$$

where $\hat{\mu}_{C,i}$ is the empirical mean of feature i for class C:

$$\hat{\mu}_{C,i} = \frac{1}{N_C} \sum_{j=1}^{N_C} x_{j,i}.$$

This result shows that the MLE for each variance component $\sigma_{C,i}^2$ is simply the sample variance.