CS 189/289A Introduction to Machine Learning
Fall 2021 Jennifer Listgarten, Jitendra Malik

Midterm

- Please do not open the exam before you are instructed to do so.
- **Electronic devices are forbidden on your person**, including cell phones, tablets, 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.
- The exam is closed book, closed notes except your one-page cheat sheet. You are allowed one double-sided handwritten 8.5x11 inch cheatsheet.
- You have 1 hour and 50 minutes (unless you are in the DSP program and have an allowance of 150% or 200% time).
- Please write your initials at the top right of each page after this one (e.g., write “JD” if you are John Doe). Finish this by the end of your 1 hour and 50 minutes.
- Mark your answers on the exam itself in the space provided. Do **not** attach any extra sheets.
- 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.
- The last question (Question 8) is for CS289A students only. Students enrolled in CS189A will **not** receive any credit for answering this question.

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

○ CS 189A

○ CS 289A

1 Multiple Choice (Single Answer)

1. (1 point) Peanut wants to train a model to accurately classify different types of animals from images. After training and testing his model, he observes that the model has high training error and high test error. What can we most confidently say about the bias/variance characteristics of Peanut's model?

   A. High bias.

   B. Low bias.

   C. High variance.

   D. Low variance.

   Solution: A. High bias because the model has high training error.

2. (1 point) Consider a binary classification data set with 9000 positively labelled examples and 1000 negatively labelled examples. What is the area under the ROC curve (AUC-ROC) of a random classifier that classifies any example as positive with probability π and as negative with probability $1 - \pi$? Here, the probability π is a hyperparameter.

   A. Close to zero.

   B. Close to 0.1.

   C. Close to 0.5.

   D. Close to 0.9.

   E. Close to one.

   Solution: C. The random classifier has a diagonal ROC curve.

3. (1 point) Again, consider a binary classification data set with 9000 positively labelled examples and 1000 negatively labelled examples. What is the precision and the recall of a classifier that always classifies any example as positive?

   A. The precision is 0.1, and the recall is 0.9.

   B. The precision is 0.9, and the recall is 0.1.

   C. The precision is 1.0, and the recall is 0.9.

   D. The precision is 0.9, and the recall is 1.0.

   E. The precision is 0.1, and the recall is 1.0.

   Solution: D. The precision is $9000/10000 = 0.9$ since all examples are classified as positive. The recall is $9000/9000 = 1.0$ (perfect recall) since all positive examples are classified as positive.

4. (2 points) Assume we are given $X \in \mathbb{R}^{n\times d}$ and $y \in \mathbb{R}^n$ for $n > d$. The Ridge regression estimator with regularization coefficient λ estimates the weight vector to be

$$\hat{w} = \arg\min_{w} \left(\left\| y - Xw \right\|_2^2 + \lambda \|w\|_2^2 \right). \tag{1}$$

The Ridge regression estimator is equivalent to the ordinary least squares estimator on which of the following modified version of X and y?

I_d denotes the $d \times d$ identity matrix. 0_d denotes the all-zero d-dimensional vector, and 1_d denotes the all-one d-dimensional vector.

A.

$$y' = \begin{bmatrix} y \\ 0_d \end{bmatrix}, X' = \begin{bmatrix} X \\ \sqrt{\lambda} I_d \end{bmatrix} \tag{2}$$

B.

$$y' = \begin{bmatrix} y \\ 1_d \end{bmatrix}, X' = \begin{bmatrix} X \\ \sqrt{\lambda} I_d \end{bmatrix} \tag{3}$$

C.

$$y' = \begin{bmatrix} y \\ 0_d \end{bmatrix}, X' = \begin{bmatrix} X \\ \lambda I_d \end{bmatrix} \tag{4}$$

D.

$$y' = \begin{bmatrix} y \\ 1_d \end{bmatrix}, X' = \begin{bmatrix} X \\ \lambda I_d \end{bmatrix} \tag{5}$$

Solution: A.

For the original X and y, the Ridge regression estimator is

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

For the modified X' and y', the ordinary least squares estimator is

$$\hat{w} = (X'^T X')^{-1} X'^T y'.$$

Therefore, we want to find X' and y' such that $X'^T X' = X^T X + \lambda I_d$ and $X'^T y' = X^T y$. For the y' and X' given in choice A,

$$X'^T X' = X^T X + (\sqrt{\lambda} I_d)^T (\sqrt{\lambda} I_d) = X^T X + \lambda I_d.$$

$$X'^T y' = X^T y + (\sqrt{\lambda} I_d)^T (0_d) = X^T y.$$

2 Multiple Choice (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** on multiple answer questions: the set of all correct answers must be checked.

1. (2 points) Which of the following statements are **TRUE** regarding positive semi-definite and positive-definite matrices?

   ○ "Every entry of a matrix is non-negative" is a necessary but insufficient condition for a matrix to be positive definite.

   ○ The singular values of a positive semi-definite matrix are the same as its eigenvalues.

   ○ If a matrix A is positive semi-definite, then there exists a matrix B such that $B^T B = A$.

   ○ The covariance matrix of any distribution is positive semi-definite and invertible.

   ○ If the Hessian of a function is positive semi-definite, then the function is convex.

   Solution: B, C, and E.

   Why A is incorrect: Positive-definite matrices can have negative entries. For example, the following matrix

   $$\begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$

   is positive-definite.

   Why B is correct: For a positive semi-definite matrix A, we can apply the spectral decomposition $A = Q\Lambda Q^{-1}$. The singular values are defined as the square roots of non-negative eigenvalues of $A^T A = Q\Lambda^2 Q^{-1}$, so the singular values are the same as eigenvalues.

   Why C is correct: If a matrix A is positive semi-definite, we can apply the spectral decomposition $A = Q\Lambda Q^{-1} = Q\Lambda Q^T$, where Λ only has non-negative diagonal entries and can be written as $\Lambda = D^2$, so $A = (QDQ^T)^T(QDQ^T)$.

   Why D is incorrect: The covariance matrix of a distribution is always positive semi-definite but could have zero determinant. For example, a deterministic distribution has zero variance.

2. (2 points) Which of the following statements are **TRUE** regarding Lasso and Ridge regression? Let $X \in \mathbb{R}^{n\times d}$ be the data matrix and $y \in \mathbb{R}^n$ be the observed labels for the n examples. Let $w \in \mathbb{R}^d$ be the weight parameters learned from Lasso or Ridge regression, and let λ be the regularization coefficient.

   ○ In the Bayesian MAP interpretation, Lasso regression can be interpreted as linear regression for which the coefficients have Poisson prior distributions.

   ○ In Lasso regression, as the regularization coefficient becomes very large ($\lambda \to \infty$), the learned weights from Lasso regression will be close to zero ($w \to 0$).

   ○ Lasso regression performs both feature selection and regularization.

   ○ There is no unique solution to Ridge regression whenever X is not full rank.

Solution: B and C.

Why A is incorrect: In the Bayesian MAP interpretation, Lasso regression can be interpreted as linear regression for which the coefficients have **Laplace** prior distributions.

Why D is incorrect: There is always a unique solution to Ridge regression when $\lambda > 0$ regardless of whether X is full rank.

3. (2 points) If the model resulting from Ridge regression is currently overfitting, what are possible things to reduce overfitting? Select all that apply.

   ○ Collect new data to increase the training data size.

   ○ Repeat the current data twice to increase the training data size.

   ○ Increase the ℓ_2 regularization penalty in the loss function.

   ○ Add new features to the model.

   ○ Remove features from the model.

Solution: A, C, and E.

4. (2 points) Which of the following statements about gradient descent are **TRUE**?

   ○ After a gradient descent update step, the objective function value at the weight vector is always lower after the update than before.

   ○ There is always a unique steepest descent direction in gradient descent.

   ○ Gradient descent converges to a globally optimal solution for logistic regression under appropriate assumptions.

   ○ Since ReLU is a convex function, a neural network that uses ReLU activations is also a convex function, and therefore gradient descent will converge to a globally optimal solution on neural networks with ReLU activations.

Solution: C only.

Why A is incorrect: If the step size is too large, the objective function value can increase after an update.

Why B is incorrect: If the gradient is zero along some dimensions, there would not be a unique steepest descent direction.

Why D is incorrect: The composition of ReLU and linear layers might not necessarily be convex. For example composing ReLU with $f(x) = -x$ results in a concave function.

5. (2 points) Which of the following statements are **TRUE** about cross entropy?

   ○ The value of cross-entropy loss is always non-negative.

   ○ Cross-entropy loss is only suitable for binary classification but not for multiclass classification.

   ○ For two discrete probability distributions P and Q, the KL divergence is symmetric, i.e. $D_{KL}(P||Q) = D_{KL}(Q||P)$.

○ Minimizing the cross-entropy is equivalent to maximizing the likelihood with respect to the model parameters.

Solution: A and D.

6. (2 points) Which of the following statements are **TRUE** about cross validation? In this problem, the final test set is **not** involved in the cross validation process.

    ○ During the k-fold cross validation process, precisely k models are trained on different subsets of the data.

    ○ During the k-fold cross validation process, precisely $k - 1$ models are trained on different subsets of the data.

    ○ During the k-fold cross validation process, we need to draw k random seeds to shuffle the data precisely k times.

    ○ At the end of the k-fold cross validation process, we choose hyperparameters that minimize the highest validation loss among the different splits.

    Solution: A only.

    Why C is incorrect: During the k-fold cross validation process, we only need to shuffle and split the data once.

    Why D is incorrect: At the end of the k-fold cross validation process, we choose hyperparameters that minimize the **average** validation loss among the different splits.

7. (2 points) Which of the following statements are **TRUE** about principle component analysis (PCA)? Given n data points of dimension d, $X \in \mathbb{R}^{n \times d}$, assume that we are performing PCA to reduce the dimension of the data to k where $k < d$. Which of the following would result in a **DIFFERENT** PCA basis? Recall that the PCA basis is a set of unit vectors.

    ○ Multiplying all columns in X by a factor of two before performing PCA.

    ○ Multiplying the first column in X by a factor of two before performing PCA.

    ○ Replacing the first column of X by the sum of all columns (including the first column).

    ○ Not subtracting the mean from X before performing PCA.

    Solution: B, C, and D.

    Why A would not result in a different PCA basis: Multiplying all columns in X by a factor of two would double the eigenvalues but would not change the basis.

8. (2 points) Which of the following statements are **TRUE** about weight updates in neural networks?

    ○ Typically, the weights in all hidden layers are initially all set to zero.

    ○ If using the mean squared error loss, the weight changes in the last layer are proportional to the difference between the model output and the true labels.

    ○ The weight changes in a particular hidden layer are proportional to the input to that weight layer.

○ Weight updates are computed in the forward pass of the network.

Solution: B and C.

Why A is incorrect: See Problem 2 in Discussion 3.

Why D is incorrect: Weight updates are computed in the **backward** pass of the network.

3 Fishing

1. (6 points) Rumble is fishing by a lake and catches n fish. She records the time she must wait in between catching each fish as x_i minutes, where x_i denotes the time she had to wait between catching fish $(i-1)$ and fish i. After catching n fish and collecting data points $x_1, \cdots, x_n$, Rumble heads home for dinner. Rumble knows that the x_i's are distributed according to a Poisson distribution, Poisson(λ), for some unknown rate $\lambda \in \mathbb{R}_+$.

   Recall that the probability mass function for a Poisson distribution with rate λ is

$$f_\lambda(x) = \frac{\lambda^x e^{-\lambda}}{x!}. \tag{6}$$

   (a) (4 points) With a prior $\lambda \sim \text{Exp}(\xi)$ for some fixed $\xi > 0$, what is the maximum a posteriori (MAP) estimate for the rate λ?

   Recall that the exponential distribution Exp(ξ) has probability density function

$$f(\lambda) = \xi e^{-\xi\lambda} \text{ for } \lambda \geq 0.$$

   Solution: The likelihood of $x_1, \cdots, x_n$ is

$$p(x_1, \cdots, x_n|\lambda) = \Pi_{i=1}^n f_\lambda(x_i) = \frac{1}{x_1!x_2!\cdots x_n!} \exp\{(\log\lambda)\sum_{i=1}^n x_i - n\lambda\}.$$

   The prior distribution of λ is

$$p(\lambda) = \xi e^{-\xi\lambda}.$$

   The posterior distribution then has log probability

$$\log p(\lambda|x_1, \cdots, x_n) = C + (\log\lambda)\sum_{i=1}^n x_i - n\lambda - \xi\lambda.$$

   The MAP estimator $\hat{\lambda}_{MAP}$ that maximizes the posterior probability satisfy the zero gradient condition

$$\frac{d}{d\lambda}\log p(\lambda|x_1, \cdots, x_n) \mid_{\hat{\lambda}=\hat{\lambda}_{MAP}} = \frac{1}{\hat{\lambda}_{MAP}}\sum_{i=1}^n x_i - n - \xi = 0. \implies \hat{\lambda}_{MAP} = \frac{1}{n+\xi}\sum_{i=1}^n x_i.$$

   (b) (2 points) In Discussion 1, we have derived that the maximum likelihood estimation of λ is the sample mean

$$\hat{\lambda}_{MLE} = \frac{1}{n}\sum_{i=1}^n x_i. \tag{7}$$

   How does the above MAP estimator compare to the MLE estimator in their bias and variance characteristics?

   A. The above MAP estimator has higher bias and lower variance.

   B. The above MAP estimator has lower bias and lower variance.

C. The above MAP estimator has higher bias and higher variance.

D. The above MAP estimator has lower bias and higher variance.

Solution: A. The two estimators are related by $\hat{\lambda}_{MAP} = \frac{n}{n+\xi}\hat{\lambda}_{MLE}$ and $\frac{n}{n+\xi} < 1$, so $\hat{\lambda}_{MAP}$ has lower variance. Since $\mathbb{E}[\hat{\lambda}_{MLE}] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[x_i] = \frac{1}{n}\sum_{i=1}^{n}\lambda = \lambda$, the MLE estimator $\hat{\lambda}_{MLE}$ is unbiased. In contrast, the MAP estimator has expected value $\mathbb{E}[\hat{\lambda}_{MAP}] = \frac{n}{n+\xi}\lambda \neq \lambda$, and is therefore biased.

4 Watermelons

1. (6 points) Finn lives on a watermelon farm and wants to classify whether a watermelon is sweet (labelled as $y = 1$) or not sweet (labelled as $y = 0$). Finn observes a d-dimensional feature vector $x \in \mathbb{R}^d$ associated with the appearance and the smell of each watermelon. Before observing a watermelon, Finn's general prior is that a watermelon is sweet with probability $p(y = 1) = \pi_1$ and not sweet with probability $p(y = 0) = 1 - \pi_1$.

   Finn is a watermelon expert and knows that the conditional probability distribution of the watermelon features $p(x|y = k)$ for class k (where $k = 0, 1$) is a d-dimensional Gaussian distribution $N(\mu_k, \mathbf{\Sigma})$ with mean $\mu_k \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{R}^{d\times d}$. Note that the same covariance matrix Σ is shared between the two classes.

   $$f(x|y = k) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\{-\frac{1}{2}(x - \mu_k)^T\Sigma^{-1}(x - \mu_k)\} \tag{8}$$

   Can you help Finn find out how likely a given watermelon is sweet? Write down a simplified expression of $p(y = 1|x)$ as a function of x, μ_0, μ_1, Σ, and π_1. The expression should be in the form of $s(w^T x + b)$ where $s(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function is the sigmoid function. Write down what w and b should be.

   Solution: By Bayes' Rule,

   $$p(y = 1|x) = \frac{p(x|y = 1)p(y = 1)}{p(x)} = \frac{p(x|y = 1)p(y = 1)}{p(x|y = 0)p(y = 0) + p(x|y = 1)p(y = 1)}.$$

   Therefore,

   $$p(y = 1|x) = \frac{\pi_1 f_1(x)}{\pi_1 f_1(x) + \pi_0 f_0(x)} = \frac{1}{1 + \frac{\pi_0 f_0(x)}{\pi_1 f_1(x)}}.$$

   When substituting in the Gaussian probability density function for $f(x)$ and writing π_1 as $e^{\log \pi_1}$,

   $$\pi_k f_k(x) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\{-\frac{1}{2}(x - \mu_k)^T\Sigma^{-1}(x - \mu_k) + \log \pi_k\} = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp(q_k(x)),$$

   where $q_k(x) = -\frac{1}{2}(x - \mu_k)^T\Sigma^{-1}(x - \mu_k) + \log \pi_k$.

   Therefore, the ratio can be written as

   $$\frac{\pi_0 f_0(x)}{\pi_1 f_1(x)} = e^{q_0(x) - q_1(x)},$$

   and hence

   $$p(y = 1|x) = \frac{1}{1 + e^{q_0(x) - q_1(x)}} = s(q_1(x) - q_0(x)).$$

   Now let's look at the expression $q_1(\mathbf{x}) - q_0(\mathbf{x})$

   $$q_1(\mathbf{x}) - q_0(\mathbf{x}) = \log\frac{\pi_1}{1 - \pi_1} - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_1)^\top\mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_1) + \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_0)^\top\mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_0)$$

$$= \log \frac{\pi_1}{1-\pi_1} + \mathbf{x}^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - \frac{1}{2}\boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_0^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0$$

Notice that we can write it out as:

$$q_1(\mathbf{x}) - q_0(\mathbf{x}) = \log \frac{\pi_1}{1-\pi_1} + \frac{1}{2}\boldsymbol{\mu}_0^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \frac{1}{2}\boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} = w_0 + \mathbf{w}^\top \mathbf{x}$$

In other words,

$$p(y = 1 \mid x) = s(w_0 + w^T x),$$

where

$$w_0 = \log \frac{\pi_1}{1-\pi_1} + \frac{1}{2}\boldsymbol{\mu}_0^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \frac{1}{2}\boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1,$$

and

$$w^T = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}^{-1}.$$

5 Road Bikes and Mountain Bikes

1. (8 points) In statistics, a mixture model is a probabilistic model for representing the presence of subpopulations within an overall population. For example, we might observe the prices of n different bikes without knowing the type of each bike (*e.g.*, road bike or mountain bike). We could model the bike prices as a mixture model with two different components, where one component corresponds to road bikes and the other corresponds to mountain bikes.

   Let us assume that the price distribution is Gaussian for each type of bikes. In this case, the overall price distribution of all bikes is a mixture model of two univariate Gaussian distributions, where the likelihood is given by

   $$p(x \mid \pi_0, \pi_1, \mu_0, \sigma_0^2, \mu_1, \sigma_1^2) = p(k = 0) f(x \mid \mu_0, \sigma_0^2) + p(k = 1) f(x \mid \mu_1, \sigma_1^2), \tag{9}$$

   where $p(k = 0) = \pi_0$ is the probability that a given bike is a road bike, $p(k = 1) = \pi_1 = 1 - \pi_0$ is the probability that a given bike is a mountain bike, and $f(x \mid \mu_k, \sigma_k^2)$ is the Gaussian probability density function with mean μ_k and variance σ_k^2.

   We observed n i.i.d. samples of bike prices $x_1, \cdots, x_n$ from this mixture of Gaussian distributions.

   (a) (2 points) What is the log probability of the bike prices $\{x_i\}_{i=1}^n$? Write down an expression for $\log p(x_1, \cdots, x_n \mid \pi_0, \pi_1, \mu_0, \sigma_0^2, \mu_1, \sigma_1^2)$.

   Solution:

   $$\begin{aligned} & \log p(x_1, \cdots, x_n \mid \pi_0, \pi_1, \mu_0, \sigma_0^2, \mu_1, \sigma_1^2) \\ = & \sum_{i=1}^{n} \log p(x_i \mid \pi_0, \pi_1, \mu_0, \sigma_0^2, \mu_1, \sigma_1^2) \\ = & \sum_{i=1}^{n} \log\{\pi_0 N(x_i \mid \mu_0, \sigma_0^2) + \pi_1 N(x_i \mid \mu_1, \sigma_1^2)\} \\ = & \sum_{i=1}^{n} \log\{\frac{\pi_0}{\sigma_0 \sqrt{2\pi}} e^{-\frac{(x_i - \mu_0)^2}{2\sigma_0^2}} + \frac{\pi_1}{\sigma_0 \sqrt{2\pi}} e^{-\frac{(x_i - \mu_0)^2}{2\sigma_0^2}}\} \end{aligned}$$

(b) (2 points) Unfortunately, there is no closed-form expression for the maximum likelihood estimator for the likelihood function above. In one approach to estimate the parameters for a Gaussian mixture model, we introduce unobserved random variables c_i associated with the component of each x_i. The unobserved binary variable $c_i \in \{0, 1\}$ indicates which of the two Gaussian distributions x_i is from (*e.g.*, bike type).

$$P(c_i = 0) = \pi_0 \text{ and } P(c_i = 1) = \pi_1. \tag{10}$$

Conditioned on the component c_i (*e.g.*, bike type), the observations $x_i \mid c_i$ (*e.g.*, bike prices) follows a Gaussian distribution $N(x_i \mid \mu_k, \sigma_k^2)$. For $k = 0, 1$,

$$p(x_i \mid c_i = k) = \frac{1}{\sigma_k \sqrt{2\pi}} \exp\{-\frac{(x-\mu_k)^2}{2\sigma_k^2}\}. \tag{11}$$

Suppose that we already know the unobserved variables $c_1, \cdots, c_n$ in addition to $x_1, \cdots, x_n$. The joint density of the samples and the unobserved components $\{(x_i, c_i)\}_{i=1}^n$ is known as the *complete likelihood*.

Find an expression for $\log p((x_1, c_1), \cdots, (x_n, c_n) \mid \pi_0, \pi_1, \mu_0, \sigma_0^2, \mu_1, \sigma_1^2)$. In your answer, you may use notations such as μ_{c_i}.

Solution:

$$\begin{aligned}
& \log p((x_1, c_1), \cdots, (x_n, c_n) \mid \pi_0, \pi_1, \mu_0, \sigma_0^2, \mu_1, \sigma_1^2) \\
= & \sum_{i=1}^n \log\{\pi_{c_i} \frac{1}{\sigma_{c_i}\sqrt{2\pi}} e^{-\frac{(x_i-\mu_{c_i})^2}{2\sigma_{c_i}^2}}\} \\
= & \sum_{i=1}^n \log \pi_{c_i} - \frac{1}{2}\log 2\pi\sigma_{c_i}^2 - \frac{(x_i-\mu_{c_i})^2}{2\sigma_{c_i}^2}
\end{aligned}$$

(c) (4 points) Still under the assumption that the unobserved component variables $c_1, \cdots, c_n$ are known, what is the maximum likelihood estimation (MLE) of μ_0, σ_0^2, μ_1, and σ_1^2? In your answer, write down expressions for $\hat{\mu}_k$ and $\hat{\sigma}_k^2$ in a way that holds for both $k = 0, 1$. Let $S_k := \{i : c_i = k \text{ for } i = 1, \cdots, n\}$ denote the set of all i's such that $c_i = k$. For example, S_0 is the set of indices for all examples from the 0-th component.

Solution:

First, let us consider the MLE estimator $\hat{\mu}_k$ for $k = 0, 1$.

$$\begin{aligned}
\frac{\partial}{\partial \mu_k} \log p((x_1, c_1), \cdots, (x_n, c_n) \mid \pi_0, \pi_1, \mu_0, \sigma_0^2, \mu_1, \sigma_1^2) &= \sum_{i=1}^{n} -\frac{\partial}{\partial \mu_k} \frac{(x_i - \mu_{c_i})^2}{2\sigma_{c_i}^2} \\
&= \sum_{i \in S_k} \frac{x_i - \mu_k}{\sigma_{c_i}^2}.
\end{aligned}$$

By solving for zero gradient, the MLE estimator $\hat{\mu}_k$ is the mean among all examples from the 0-th component:

$$\hat{\mu}_k = \frac{1}{|S_k|} \sum_{i \in S_k} x_i.$$

For the MLE estimator of σ_k^2, the partial derivative w.r.t. σ_k^2 is

$$\begin{aligned}
\frac{\partial}{\partial \sigma_k^2} \log p((x_1, c_1), \cdots, (x_n, c_n) \mid \pi_0, \pi_1, \mu_0, \sigma_0^2, \mu_1, \sigma_1^2) &= \sum_{i=1}^{n} -\frac{\partial}{\partial \sigma_k^2} (\frac{1}{2} \log \sigma_{c_i}^2) - \frac{\partial}{\partial \sigma_0^2} \frac{(x_i - \mu_{c_i})^2}{2\sigma_{c_i}^2} \\
&= \sum_{i \in S_k} -\frac{1}{2\sigma_k^2} + \frac{(x_i - \mu_k)^2}{2\sigma_k^4}.
\end{aligned}$$

By solving for zero gradient, the MLE estimator $\hat{\sigma}_k^2$ is

$$\hat{\sigma}_k^2 = \frac{1}{|S_k|} \sum_{i \in S_k} (x_i - \hat{\mu}_k)^2.$$

This is the maximization (M) step of the expectation-maximization (EM) algorithm for estimating the parameters of a Gaussian mixture model.

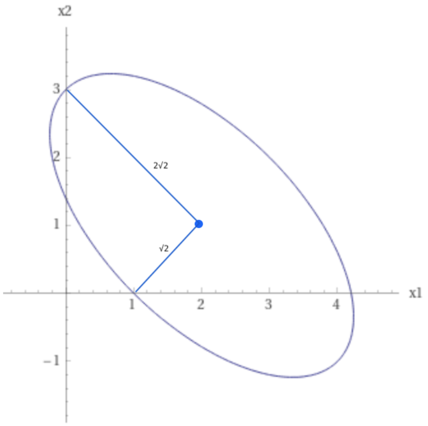
6 Level Sets

1. (7 points) Esme generated a 2D multivariate Gaussian distribution but lost the parameters used for generating the multivariate Gaussian distribution. The only piece of paper left has a plot of the level set of the probability density function f at value

$$f(x_1, x_2) = \frac{1}{e}(\text{max Gaussian density}).$$

Here, "max Gaussian density" refers to the maximum value of the Gaussian probability density function. A picture of the level set is shown below. For a function $f : \mathbb{R}^2 \to \mathbb{R}$ and a constant $c \in \mathbb{R}$, the level set is defined as $\{x \in \mathbb{R}^2 : f(x) = c\}$.

The level set on Esme's paper is an ellipse centered at $(2, 1)$ with axes along the unit vectors $(1, -1)$ and $(1, 1)$. The two axes correspondingly have radii $r_1 = 2\sqrt{2}$ and $r_2 = \sqrt{2}$ in the ellipse.



(a) (1 point) What is the mean of the multivariate Gaussian distribution?

Solution: The level set is centered at the mean of the multivariate Gaussian distribution. Therefore, $\mu = (2, 1)$.

(b) (1 point) Suppose we have a 2D Gaussian distribution with zero mean and diagonal covariance matrix

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}.$$

What is the maximum density in this distribution?

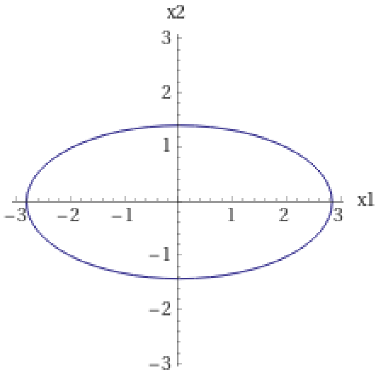
Solution: The density is given by

$$f(x) = \frac{1}{2\pi\sigma_1\sigma_2}\exp\{-\frac{x_1^2}{2\sigma_1^2} - \frac{x_2^2}{2\sigma_2^2}\}$$

The maximum density occurs at zero, so the maximum density is

$$\max_x f(x) = f(0) = \frac{1}{2\pi\sigma_1\sigma_2}.$$

(c) (2 points) Consider a rotated version of the ellipse that is aligned with the two axes and centered at zero (as illustrated below). As before, the radii of the ellipse are $2\sqrt{2}$ and $\sqrt{2}$. What is the multivariate Gaussian distribution that has the rotated ellipse as its level set at $1/e$ of the maximum density?



Recall that the equation of an ellipse centered at zero and aligned with the x_1- and x_2- axis radii a and b is

$$\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} = 1. \tag{12}$$

Solution: If we center the ellipse and rotate the ellipse 45 degrees counter-clockwise to align with the x-axis, we would have an ellipse with equation

$$\frac{x_1^2}{8} + \frac{x_2^2}{2} = 1. \tag{13}$$

The axis-aligned ellipse corresponds to the level set of a diagonal Gaussian distribution with zero mean and diagonal covariance matrix

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}.$$

Since the level set occurs at $f(x_1, x_2) = \frac{1}{e} \cdot \frac{1}{2\pi\sigma_1\sigma_2}$, it corresponds to all points that satisfy the equation

$$f(x) = \frac{1}{2\pi\sigma_1\sigma_2}\exp\{-\frac{x_1^2}{2\sigma_1^2} - \frac{x_2^2}{2\sigma_2^2}\} = \frac{1}{e} \cdot \frac{1}{2\pi\sigma_1\sigma_2}$$

$$\implies \exp\{-\frac{x_1^2}{2\sigma_1^2} - \frac{x_2^2}{2\sigma_2^2}\} = \frac{1}{e}$$

$$\implies \frac{x_1^2}{2\sigma_1^2} + \frac{x_2^2}{2\sigma_2^2} = 1.$$

Therefore, by matching this to the axis-aligned ellipse equation, we arrive at

$$\sigma_1 = 2 \text{ and } \sigma_2 = 1.$$

(d) (3 points) What is the covariance matrix of the multivariate Gaussian distribution for the level set described in the beginning of the question? It is sufficient to write the covariance matrix in the form of $\Sigma = QDQ^T$.

Hint: The determinant of the covariance matrix Σ is the product of its eigenvalues. Think also about the spectral decomposition of Σ.

Solution: The spectral decomposition factorizes Σ into $\Sigma = QDQ^T$, where D is diagonal and Q is an orthogonal matrix. The columns of Q are eigenvectors of Σ.

Geometrically, we know that the eigenvectors of Σ are along the directions of the two axes of symmetry in the ellipse, namely $(1, 1)$ and $(1, -1)$. Since the eigenvectors are up to the sign, equivalently the eigenvectors can also be chosen to be $(-1, -1)$ and $(-1, 1)$.

Therefore, Q is the rotation matrix (up to multiplying each column by -1)

$$Q = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}.$$

Intuitively, this rotation matrix corresponds to the 45-degree clockwise rotation. It rotates the axis-aligned eclipse back to the original tilted ellipse.

The diagonal matrix D corresponds to the eigenvalues of Σ, which in this case are the variance along the directions of the two axes of symmetry in the ellipse:

$$D = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}.$$

Putting it together, the covariance matrix is $\Sigma = QDQ^T$.

7 Activation Functions

1. (6 points) While Rectified Linear Unit (ReLU) is often the default activation function used across the deep learning community, other activation functions have also been proposed to replace ReLU. Consider the following activation function (known as *Swish* or SiLU):

$$f(x) = x \cdot \sigma(\beta x), \tag{14}$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function and β is a fixed hyperparameter.

Like ReLU, this activation function is unbounded above and bounded below. Unlike ReLU, this activation function is smooth and non-monotonic.

(a) (1 point) What is the derivative $f'(x)$ of this new activation function (for a fixed β)?

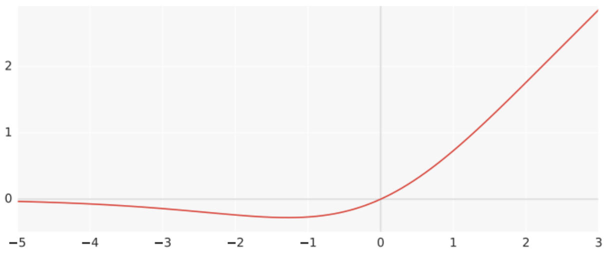
Solution:

$$f'(x) = \sigma(\beta x) + \beta x \sigma(\beta x)(1 - \sigma(\beta x)).$$

(b) (2 points) What does this activation function look like for $\beta = 1$ and for $\beta \to \infty$? Draw qualitative pictures for both values of β.

Solution: When $\beta \to \infty$, $f(x) = x \cdot \sigma(\beta x)$ becomes the ReLU function. This is because for $x > 0$, we have $\beta x \to \infty$ and $\sigma(\beta x) \to 1$, so $f(x) \to x$. In the meantime, for $x < 0$, $\beta x \to -\infty$ and $\sigma(\beta x) \to 0$, so $f(x) \to 0$. Therefore, as $\beta \to \infty$, $f(x)$ converges to the ReLU function.

Below is the picture for $\beta = 1$.



Consider a binary classification data set of 20000 examples. Among the 20000 examples, 5000 of them have the positive class label, and the remaining 15000 have negative class labels. Suppose we have a fully-connected neural network with one hidden layer and with sigmoid activation after the output layer. When training this neural network for one epoch using stochastic gradient descent (SGD), we forget to shuffle the examples and loop through all the negative examples first before the positive examples.

(c) (1 point) After the model is trained on the 15000 examples with negative class labels, which of the following is likely true?

A. The model output **before** the sigmoid activation would always be close to zero.

B. The model output **before** the sigmoid activation would always be very negative.

C. The model output **before** the sigmoid activation would always be very positive.

D. The model output **before** the sigmoid activation would sometimes be positive and sometimes be negative.

Solution: B.

(d) (1 point) Suppose we use ReLU as the activation function in the hidden layer. In the above scenario, after training on only negatively-labelled examples, the activation input to the ReLU function might end up being always negative. What would then happen when the model is trained on the positive-labelled examples afterwards? Describe qualitatively in 1-2 sentences.

Solution: When the activation input to ReLU is negative, the ReLU function is flat and no gradients would go through the ReLU function. As all gradients would be zero, the model would be stuck even after seeing positive examples.

(e) (1 point) Answer the above question with $f(x) = x \cdot \sigma(\beta x)$ as the activation function with $\beta = 1$.

Solution: With the new activation function, there would be non-zero gradients and the model weights will update after seeing positive examples.

8 Projected Gradient Descent for Lasso Regression (CS 289A Only)

1. (6 points) Sometimes, we may wish to add constraints to an optimization problem. For example, we might want certain variables to be non-negative. Or, we might need to satisfy budget constraints and resource constraints.

   In this problem, we will consider Lasso regression. Let $X \in \mathbb{R}^{n \times d}$ be the features and $y \in \mathbb{R}^n$ be the labels of a regression problem. Given a regularization coefficient λ, the objective of Lasso regression is to solve

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \left\| y - Xw \right\|_2^2 + \lambda \|w\|_1. \tag{15}$$

   Unlike Ridge regression, the loss function for Lasso regression is not differentiable. One way to tackle the optimization problem for Lasso is to rewrite the above objective into the following smooth, constrained optimization problem (by separating w into $w = w^+ - w^-$ for two non-negative vectors w^+ and w^- representing the positive and negative parts of w):

$$\min_{w^+, w^- \in \mathbb{R}^d} \frac{1}{n} \left\| y - X(w^+ - w^-) \right\|_2^2 + \lambda \mathbb{1}_d^T w^+ + \lambda \mathbb{1}_d^T w^- \tag{16}$$

$$\text{subject to: } w_i^+, w_i^- \geq 0 \text{ for } i = 1, \cdots, d.$$

   $\mathbb{1}_d$ denotes the all-one d-dimensional vector and therefore $\mathbb{1}_d^T w^+ = \sum_{i=1}^d w_i^+$.

   In general, a constrained minimization problem for a given constraint set $C \subset \mathbb{R}^d$ asks for the best solution $\min_{w \in C} f(w)$ inside the set C. In the above constrained formulation of Lasso regression, if we concatenate w_i^+ and w_i^- together as a new variable $u = [w_i^+ \; w_i^-] \in \mathbb{R}^{2d}$, the constraint set is the positive orthant in $\mathbb{R}^{2d}$

$$C = \{u \in \mathbb{R}^{2d} : u_i \geq 0 \text{ for } i = 1, \cdots, 2d\}. \tag{17}$$

   We have seen that gradient descent is a standard way to solve unconstrained optimization problem. Here, we will see that we can modify gradient descent into projected gradient descent to solve constrained optimization problems.

   (a) (2 points) Ignoring the constraint for a moment, what is one step of the gradient descent update for w^+ and w^- in the following unconstrained regression problem?

$$\min_{w^+, w^- \in \mathbb{R}^d} \frac{1}{n} \left\| y - Xw^+ + Xw^- \right\|_2^2 + \lambda \mathbb{1}_d^T w^+ + \lambda \mathbb{1}_d^T w^- \tag{18}$$

   Assume that the step size is η.

Solution:

$$w_{k+1}^{+} = w_k^{+} - \eta\left(\frac{2}{n}[X^T X(w_k^{+} - w_k^{-}) - X^T y] + \lambda \mathbb{1}_d\right)$$

$$w_{k+1}^{-} = w_k^{-} - \eta\left(\frac{2}{n}[X^T y - X^T X(w_k^{+} - w_k^{-})] + \lambda \mathbb{1}_d\right)$$

(b) (2 points) Now back to the constrained optimization problem. When C is a convex set, the projection operator $P_C : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ onto a convex set C is defined for $a \in \mathbb{R}^{2d}$ as

$$P_C(a) := \underset{b \in C}{\arg\min} \|a - b\|_2^2. \tag{19}$$

The projection onto a convex set is uniquely defined. Give an expression (without proof) for the projection P_C onto the positive orthant

$$C = \{a \in \mathbb{R}^{2d} : a_i \geq 0 \text{ for } i = 1, \cdots, 2d\}. \tag{20}$$

Solution:

$$P_C(a) = \max(0, a),$$

where the max is elementwise for the vector.

(c) (2 points) Projected gradient descent introduces an additional projection in every gradient descent update, which aims to minimize loss while guaranteeing that the weight stays within the constraint set C. For the constraint set $C \subset \mathbb{R}^{2d}$, starting from a initial point $u_0 \in C$, with stepsize $\eta > 0$, projected gradient descent iterates the following equation until a stopping condition is met:

$$u_{k+1} = P_C(u_k - \eta \nabla f(u_k)). \tag{21}$$

For the reformulated Lasso optimization problem in Equation 16, Write down an expression for the projected gradient descent update for w^+_{k+1}, w^-_{k+1} as a function of w^+_k and w^-_k, X, Y, λ, and η.

Solution:

$$w^+_{k+1} = \max\left(0, w^+_k - \eta\left(\frac{2}{n}[X^T X(w^+_k - w^-_k) - X^T y] + \lambda \mathbb{1}_d\right)\right)$$
$$w^-_{k+1} = \max\left(w^-_k - \eta\left(\frac{2}{n}[X^T y - X^T X(w^+_k - w^-_k)] + \lambda \mathbb{1}_d\right)\right)$$