CS 189/289A Introduction to Machine Learning
Fall 2022 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. Leave your cell phone off and in a bag; it should not be visible during the exam.
- The exam is closed book and closed notes except for your one-page 8.5×11 inch cheat sheet.
- You have 1 hour and 50 minutes (unless you are in the DSP program and have a larger time allowance).
- 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 choice questions, fill in the bubble for the single best choice.
- For short and long answer questions, write within the boxes provided.
- **The last question (Question 7) is for CS289A students only**. Students enrolled in CS189 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 189

○ CS 289A

This page intentionally left blank.

initial here

1 Multiple Choice

For the following questions, select the **single best response**. Each question is worth 1.5 points.

1. Assume a linear model $Y = Xw^* + z$, where $z \sim \mathcal{N}(0, I_n)$ and w^* is the true parameter we are trying to estimate. Consider the following objective:

$$\hat{w} = \text{argmin}_{w \in \mathbb{R}^d} \|Xw - Y\|_2^2 + \lambda \|w\|_2^2 \qquad \lambda > 0 \tag{1}$$

How will increasing λ in Equation 1 affect the bias of the resulting estimator $\hat{w}$?

- ◯ Bias will increase.
- ◯ Bias will decrease.
- ◯ Bias will remain unchanged.

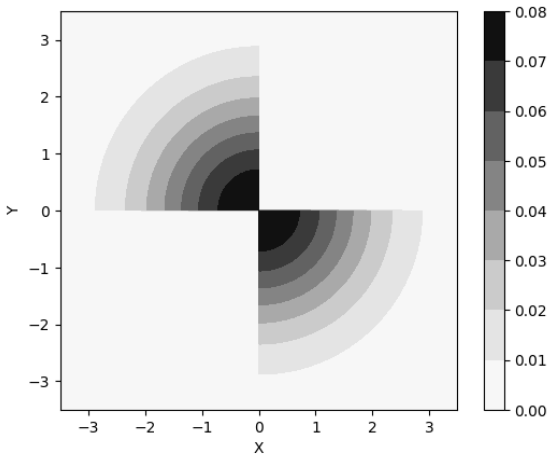
Solution: (a) Bias will increase.

2. How will increasing λ in Equation 1 affect the variance of the resulting estimator $\hat{w}$?

- ◯ Variance will increase.
- ◯ Variance will decrease.
- ◯ Variance will remain unchanged.

Solution: (b) Variance will decrease.

3. Consider the following probability density function:



Select the best description.

- ◯ X and Y appear to be both marginally and jointly Gaussian.
- ◯ X and Y appear to be marginally Gaussian but not jointly Gaussian.
- ◯ X and Y appear to be jointly Gaussian but not marginally Gaussian.

Solution: (b) X and Y appear to be marginally Gaussian but not jointly Gaussian.

initial here

4. For this question, $\mathbf{X} \in \mathbb{R}^{n \times d}$ denotes an input data matrix of rank d, $\mathbf{y} \in \mathbb{R}^n$ an outcome vector, $\mathbf{w}_{\text{ridge}}$ the ridge regression solution, and $\mathbf{w}_{\text{OLS}}$ the solution to unregularized linear regression. Select the **false** statement.

   ◯ For all $\mathbf{X}$ and true linear predictors $\mathbf{w}^* \in \mathbb{R}^d$, under the statistical assumption $\mathbf{y} = \mathbf{X}\mathbf{w}^* + \mathbf{z}$, $\mathbf{z} \sim N(0, \sigma^2 I_n)$, we have $\text{bias}(\mathbf{w}_{\text{ridge}}) \geq \text{bias}(\mathbf{w}_{\text{OLS}})$, where the bias of an estimate $\hat{\mathbf{w}}$ of $\mathbf{w}^*$ is defined as $\text{bias}(\hat{\mathbf{w}}) = \|\mathbb{E}[\hat{\mathbf{w}}] - \mathbf{w}^*\|_2$.

   ◯ For all $\mathbf{X}$ and $\mathbf{y}$, $\|\mathbf{w}_{\text{ridge}}\|_2 \leq \|\mathbf{w}_{\text{OLS}}\|_2$.

   ◯ For all $\mathbf{X}$ and $\mathbf{y}$, $\|\mathbf{w}_{\text{ridge}}\|_1 \leq \|\mathbf{w}_{\text{OLS}}\|_1$.

   Solution: The false statement is (c).

   On the exam, the way to solve this is by elimination. A is true because OLS is unbiased. B is true because w_{ridge} is a $\|\cdot\|_2$-norm penalized version of w_{OLS}. C a priori may appear to be false because we are not explicitly penalizing the ℓ_1 norm, but to confirm it is false, a counterexample is needed.

   For our counterexample, we choose $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top$ where $\mathbf{U}$ is any orthogonal 2×2 matrix, and

$$\Sigma^2 = \begin{bmatrix} 100 & 0 \\ 0 & 0.001 \end{bmatrix}, \qquad \mathbf{V} = \begin{bmatrix} \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{2}} \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{bmatrix} \tag{2}$$

   We choose $\mathbf{y} = \mathbf{U}\Sigma^{-1}\begin{bmatrix} 100 \\ 0.0005 \end{bmatrix}$. Therefore, choosing ridge with $\lambda = 1$, we have

$$\begin{aligned} w_{\text{OLS}} &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} = \mathbf{V}\Sigma^{-2}\mathbf{V}^\top\mathbf{V}\Sigma\mathbf{U}^\top\mathbf{U}\Sigma^{-1}\begin{bmatrix} 100 \\ 0.0005 \end{bmatrix} \\ &= V\Sigma^{-2}\begin{bmatrix} 100 \\ 0.0005 \end{bmatrix} = V\begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{2.5}{\sqrt{5}} \end{bmatrix} \\ w_{\text{ridge}} &= (\mathbf{X}^\top\mathbf{X} + I)^{-1}\mathbf{X}^\top\mathbf{y} = \mathbf{V}(\Sigma^2 + I)^{-1}\mathbf{V}^\top\mathbf{V}\Sigma\mathbf{U}^\top\mathbf{U}\Sigma^{-1}\begin{bmatrix} 100 \\ 0.0005 \end{bmatrix} \\ &= \mathbf{V}(\Sigma^2 + I)^{-1}\begin{bmatrix} 100 \\ 0.0005 \end{bmatrix} = \mathbf{V}\begin{bmatrix} \frac{1}{101} & 0 \\ 0 & \frac{1}{1.001} \end{bmatrix}\begin{bmatrix} 100 \\ 0.0005 \end{bmatrix} = \mathbf{V}\begin{bmatrix} \frac{100}{101} \\ \frac{0.0005}{1.001} \end{bmatrix} \\ &\approx \mathbf{V}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix}. \end{aligned}$$

   Thus we have $\|w_{\text{OLS}}\|_1 = \frac{2.5}{\sqrt{5}}$ while $\|w_{\text{ridge}}\|_1 \approx \frac{3}{\sqrt{5}}$.

5. Suppose you have a dataset where the label is binary and generated by a fair coin toss and the input features are generated by sampling i.i.d. from a standard Gaussian, independently of the label. Let n denote the number of examples in your dataset and d the number of input features. You perform logistic regression using a random 70/30 train-validation split. Let $\text{Acc}_{\text{train}}$ denote the training accuracy and Acc_{val} the validation accuracy. Select the **false** statement.

   ◯ As $n \to \infty$, $\text{Acc}_{\text{train}}$ approaches 50%.

   ◯ As $n \to \infty$, Acc_{val} approaches 50%.

initial here

◯ As $d \to \infty$, $\text{Acc}_{\text{train}}$ approaches 100%.

◯ As $d \to \infty$, Acc_{val} approaches 100%.

Solution: (d) is the false statement. The labels on the validation set are independent of the predictions made on the validation set, because those predictions are a function of the training set. Thus, the validation accuracy is always a $\text{Binom}(n, 0.5)$ random variable divided by n times 100%.

A can be seen to be true intuitively because as we get more samples while the complexity of our model stays the same, train performance should become more similar to true performance. B is true because the validation accuracy always has mean 50%, and if we have a bigger validation set, the mean should converge to its mean. C is true because once we have enough features the range of X becomes all of $\mathbb{R}^n$ and we perfectly fit the training data.

6. You are walking down Shattuck Ave. when you find a quarter on the ground. You see nothing unusual about this quarter, so you figure it is almost certainly a fair coin, though you realize that manufacturing irregularities in the coin minting process mean that coins are rarely *exactly* fair. You toss the coin 10 times and observe the following outcomes:

H H H H H H H H H T

with H denoting heads and T denoting tails. Assume coin tosses are independent. What is the maximum likelihood estimate of the next toss being heads?

◯ $\frac{5}{10}$

◯ between $\frac{5}{10}$ and $\frac{9}{10}$

◯ $\frac{9}{10}$

◯ more than $\frac{9}{10}$

Solution: (c) $\frac{9}{10}$. The MLE does not take into account the prior.

7. Consider the setup of the previous problem (Problem 6). What is the maximum a posteriori (MAP) estimate of the next toss being heads?

◯ $\frac{5}{10}$

◯ between $\frac{5}{10}$ and $\frac{9}{10}$

◯ $\frac{9}{10}$

◯ more than $\frac{9}{10}$

Solution: (b) between $\frac{5}{10}$ and $\frac{9}{10}$. The MAP estimate takes into account the prior.

8. Consider a binary classification problem with two outcomes: positive and negative. The F_1 score of a classifier in such a problem is the harmonic mean between its precision and recall:

$$F_1 = 2\frac{p \cdot r}{p + r}.$$

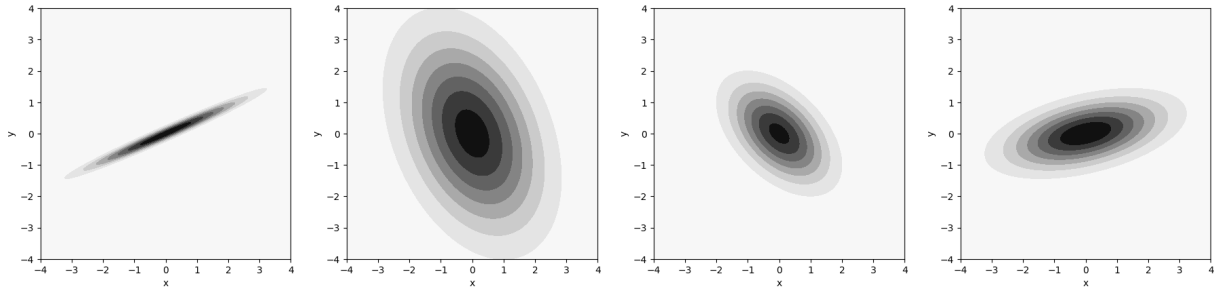
Select the **true** statement relating a classifier's F_1 score on the test set to the bias and variance of its estimated parameters. You may assume the classifier is operating in the classical (non-interpolating) regime of bias and variance.

initial here

- ◯ Low bias implies a high F_1 score.
- ◯ Low variance implies a high F_1 score.
- ◯ A low F_1 score implies a high variance.
- ◯ A high F_1 score implies low bias.

Solution: (d) A high F_1 score implies low bias. In order to have a high F1 score, the overall classification error must be low, which implies that the bias of the classifier is low. Note that a low bias does *not* imply a high F_1 score. For example, if a classification problem had 95% of its outcomes as negative, and only 5% as positive, a classifier could achieve a low bias by classifying every datapoint as negative. However, this classifier will have a low F_1 score (of 0, in fact), given that it is not able to correctly classify any true positives.

9. Which of the following could depict the probability density function of a multivariate Gaussian with covariance matrix $\Sigma = \begin{bmatrix} \sigma_X^2 & \text{cov}(X,Y) \\ \text{cov}(Y,X) & \sigma_Y^2 \end{bmatrix} = \begin{bmatrix} 2.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$?



(a) (b) (c) (d)

- ◯ (a)
- ◯ (b)
- ◯ (c)
- ◯ (d)

Solution: (d)

10. Julia is using SNE to visualize her high-dimensional dataset in two dimensions. She runs her code twice to find that it outputs different visualizations each time. Why is this expected? You may assume that Julia did not intentionally make her code deterministic by, for example, fixing the random seed.

- ◯ This is expected due to the nonconvexity of the optimization objective, and also occurs with t-SNE.
- ◯ This is expected due to the Gaussian distributions in SNE, and could be addressed by using t-SNE instead.
- ◯ This is expected due to the iterative nature of SNE, and could be addressed by using PCA to find the solution to the SNE objective without gradient descent.

Solution: (a) This is expected due to the nonconvexity of the optimization objective, and also occurs with t-SNE

initial here

2 Short Answer

1. (2 points) Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ denote a mean-centered data matrix. with an SVD of $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where the singular values are ordered in $\boldsymbol{\Sigma}$ left to right by size $\sigma_1 \geq \sigma_2 \geq \ldots$. Write the rank-$k$ PCA reconstruction of $\boldsymbol{X}$. You may denote the submatrix consisting of the first k columns of a matrix $\mathbf{M}$ as $\mathbf{M}_k$.

Solution: $\mathbf{X}\mathbf{V}_k\mathbf{V}_k^\top$. Equivalently, $\mathbf{U}_k\Sigma_k V_k^\top$.

2. (2 points) Consider the following conditional distributions of X, a discrete random variable taking values in $\{0, 1, 2, 3, 4, 5\}$, given a binary label Y:

X	0	1	2	3	4	5
$P(X\|Y = 1)$	0.1	0.1	0.1	0.2	0.2	0.3
$P(X\|Y = 0)$	0.3	0.2	0.2	0.1	0.1	0.1

If the prior label probabilities are $P(Y = 0) = P(Y = 1) = \frac{1}{2}$, calculate the Bayes risk, i.e. the minimum probability of misclassification by a classifier taking X as input.

Solution: Because priors are equal, the Bayes classifier $f^*(x)$ will choose i that maximizes $P(x|Y = i)$. Then the Bayes risk is simply

$$\sum_{i=0}^{1} P(Y = i) \sum_{x=0}^{5} P(x|Y = i)1[f^*(x) \neq i] = 0.5(0.1 + 0.1 + 0.1) + 0.5(0.1 + 0.1 + 0.1) = 0.3$$

3. (2 points) Big Bird is training a five-layer neural network with ReLU activations to classify MNIST digits. His current network achieves a 90% validation accuracy. He wants to try different activation functions to try to improve his network. Will the validation accuracy increase or decrease when he replaces the ReLU function with the identity function? **Justify your answer in a short sentence.**

Solution: The validation accuracy will decrease because now the neural network is equivalent to linear regression.

initial here

4. (2 points) Consider $\mathbf{w}_{\text{OLS}}$, the maximum likelihood estimate for $\mathbf{w}$ in the model

$$\mathbf{y} = \boldsymbol{X}\mathbf{w} + \mathbf{z} \quad \mathbf{z} \sim \mathcal{N}(0, \sigma^2 I).$$

How does $\mathbf{w}_{\text{OLS}}$ compare to the MAP estimate $\mathbf{w}_{\text{MAP}}$ if you additionally assume a Gaussian prior $\mathcal{N}(0, \sigma^2 I)$ on $\mathbf{w}$?

Solution: The $L2$ norm of $\mathbf{w}_{\text{OLS}}$ will be greater than that of $\mathbf{w}_{\text{MAP}}$.

5. (2 points) Consider once again the MAP estimate $\mathbf{w}_{\text{MAP}}$ under the model of the previous question and the same Gaussian prior $\mathcal{N}(0, \sigma^2 I)$. If the i^{th} dimension of the estimate $\mathbf{w}_{\text{MAP}}$ is equal to 0:

$$(\mathbf{w}_{\text{MAP}})_i = 0,$$

what must be true of $\boldsymbol{X}$ and $\mathbf{y}$? Assume that each feature of the data has mean 0 and that the data have been whitened such that $\boldsymbol{X}^\top \boldsymbol{X} = n\boldsymbol{I}$.

Solution: $y^\top X_i = 0$

6. (2 points) In class and in your homework, you learned about the Rectified Linear Unit (ReLU) activation function. A related activation function is the Gaussian Error Linear Unit (GELU) activation function, defined as:

$$G(x) = x\mathbb{P}(X \leq x) = x \int_{-\infty}^{x} p(z)d(z)$$

in which $X \sim \mathcal{N}(0, 1)$, $\mathbb{P}(X \leq x)$ is the CDF of a standard Gaussian, and $p(x)$ is the PDF of the standard Gaussian. Find the derivative of the GELU function $\frac{dG(x)}{dx}$ in terms of x, X, p, and $\mathbb{P}$.

Solution:

$$\begin{aligned}\frac{\partial G(x)}{\partial x} &= \frac{\partial x}{\partial x}\mathbb{P}(X \leq x) + x\frac{\partial \mathbb{P}(X \leq x)}{\partial x} \\ &= \mathbb{P}(X \leq x) + xp(x)\end{aligned}$$

initial here

7. (2 points) Consider a Gaussian covariance matrix $\Sigma \in \mathbb{R}^{2\times 2}$ with the following eigenvectors:

$$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \text{ with eigenvalue 4, and } \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \text{ with eigenvalue 2.}$$

Write Σ as a 2×2 matrix.

Solution:

$$\begin{aligned} Q\Lambda Q^\top &= \left(\frac{1}{\sqrt{2}}\right)^2 \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 6 & 2 \\ 2 & 6 \end{bmatrix} \\ &= \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \end{aligned}$$

8. (2 points) Consider a covariance matrix Σ with the same eigenvectors and eigenvalues as described in the previous question. Find the square root $\Sigma^{\frac{1}{2}}$ of Σ, once again simplifying to a 2×2 matrix.

Solution:

$$\begin{aligned} \Sigma^{\frac{1}{2}} &= Q\Lambda^{\frac{1}{2}}Q^\top \\ &= \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{4} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 2 & -\sqrt{2} \\ 2 & \sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 2+\sqrt{2} & 2-\sqrt{2} \\ 2-\sqrt{2} & 2+\sqrt{2} \end{bmatrix} \end{aligned}$$

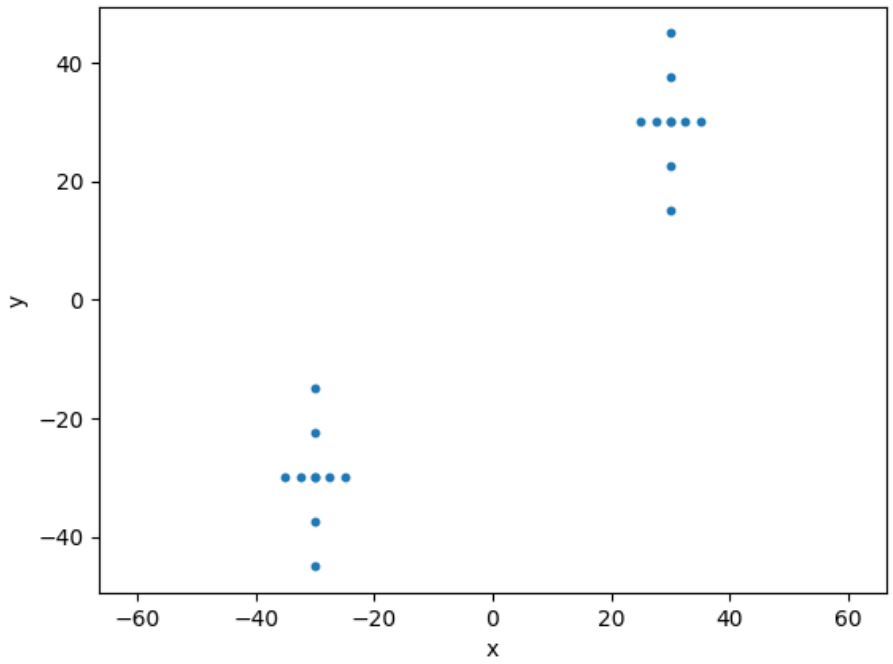
9. (2 points) Gina is training a neural network with two hidden layers to classify MNIST digits. Each image has $28 \times 28 = 784$ features. The two hidden layers of the network both have 100 hidden units, and the output is one-dimensional. All layers (including the output) have a bias term. How many learnable parameters does the neural network have?

Solution: $\underbrace{784 \cdot 100 + 100}_{\text{layer 1}} + \underbrace{100^2 + 100}_{\text{layer 2}} + \underbrace{101}_{\text{output layer}} = 88701$

initial here

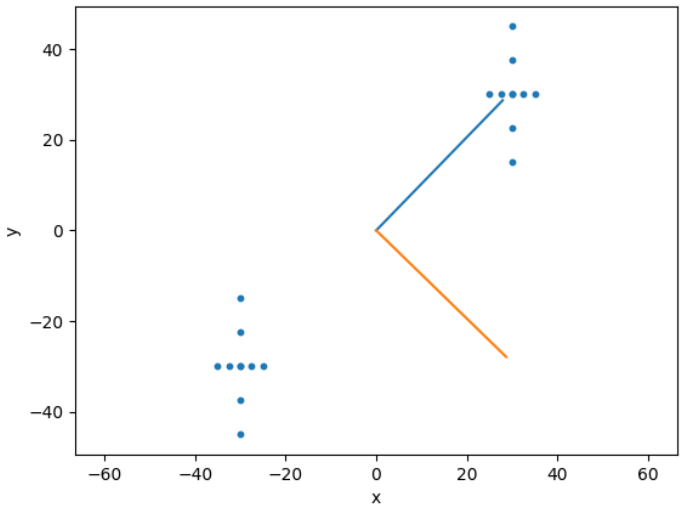
10. (2 points) Consider the following datapoints in $\mathbb{R}^2$:

    On the graph below, draw the two principal directions that PCA would return if run on the datapoints. Indicate which component explains more variance.



Solution:

The first principal component (in blue below) should be the close to the $y = x$ line. The second principal component should be orthogonal to it, so should be close to the $y = -x$ line. The first component explains more variance.



initial here

3 Fishing with a MAP

Oski is fishing for salmon. There are many types of fish in the great outdoors, so Oski is interested in the number of fish he must catch before finding a salmon. Oski realizes that he can model this as a geometric random variable

$$p(x \mid \theta) = (1-\theta)^{x-1}\theta,$$

in which θ is the underlying probability that a fish is a salmon, and $x-1$ is the number of fish Oski catches before his first salmon.

(a) (2 points) Given a dataset of n trials $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$, write down the log likelihood function $\log p(\mathcal{D} \mid \theta)$. For the rest of this problem, you may assume that any stationary point of the log likelihood function is the global maximum.

Solution:

$$\begin{aligned}
\log p(\mathcal{D} \mid \theta) &= \log \prod_{i=1}^{n} p(x_i \mid \theta) \\
&= \sum_{i=1}^{n} \log p(x_i \mid \theta) \\
&= \sum_{i=1}^{n} \log \left[(1-\theta)^{x_i-1}\theta\right] \\
&= \sum_{i=1}^{n} \left[(x_i-1)\log(1-\theta) + \log\theta\right] \\
&= n\log\theta + \log(1-\theta)\left[\sum_{i=1}^{n} x_i - n\right]
\end{aligned}$$

initial here

(b) (1 point) Suppose Oski only has a single datapoint: $x_1 = 4$. What is the maximum likelihood estimate $\hat{\theta}_{\text{MLE}}$?

Solution:

One valid approach is to skip to the next part by writing θ_{MLE} in terms of an entire dataset, then plugging in the single-entry dataset $\mathcal{D} = \{x_i\}$.

$$\begin{aligned}
\log p(4 \mid \theta) &= \log\left[(1-\theta)^3\theta\right] \\
&= 3\log(1-\theta) + \log\theta \\
\frac{\partial \log p(4 \mid \theta}{\partial\theta} &= -\frac{3}{1-\theta} + \frac{1}{\theta}
\end{aligned}$$

Setting the derivative equal to 0 yields:

$$\begin{aligned}
\frac{1}{\theta} &= \frac{3}{1-\theta} \\
\Rightarrow \theta_{\text{MLE}} &= \frac{1}{4}
\end{aligned}$$

initial here

(c) (2 points) Now suppose that Oski has collected many salmon, and has many datapoints $\{x_i\}_{i=1}^n$. Write an expression for the maximum likelihood estimate, $\hat{\theta}_{MLE}$, as a function of the datapoints.

Solution:

$$\frac{\partial \log p(\mathcal{D} \mid \theta)}{\partial \theta} = \frac{n}{\theta} - \frac{\sum_{i=1}^n x_i - n}{1-\theta}$$

Setting the derivative equal to 0 yields:

$$\begin{aligned} \frac{n}{\theta} &= \frac{\sum_{i=1}^n x_i - n}{1-\theta} \\ \Rightarrow n - n\theta &= \theta \sum_{i=1}^n x_i - n\theta \\ \Rightarrow \theta_{\text{MLE}} &= \frac{n}{\sum_{i=1}^n x_i} \end{aligned}$$

(d) (3 points) Oski realizes that a maximum likelihood estimate of θ might have been unreliable when he only had a single data point. He considers instead using maximum a posteriori (MAP) estimation by putting a prior $p(\theta)$ on θ. He settles on a Beta(α, β) distribution:

$$p(\theta) = \text{Beta}(\alpha,\beta) = \frac{1}{B(\alpha,\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1},$$

in which α and β are constants and $B(\alpha,\beta)$ is a normalizing constant. Show that the posterior distribution over parameters $p(\theta \mid x)$ is Beta(γ,δ), writing γ and δ in terms of α, β, θ and a single datapoint x. (Do not plug in $x = 4$ as in part (b).)

Solution:

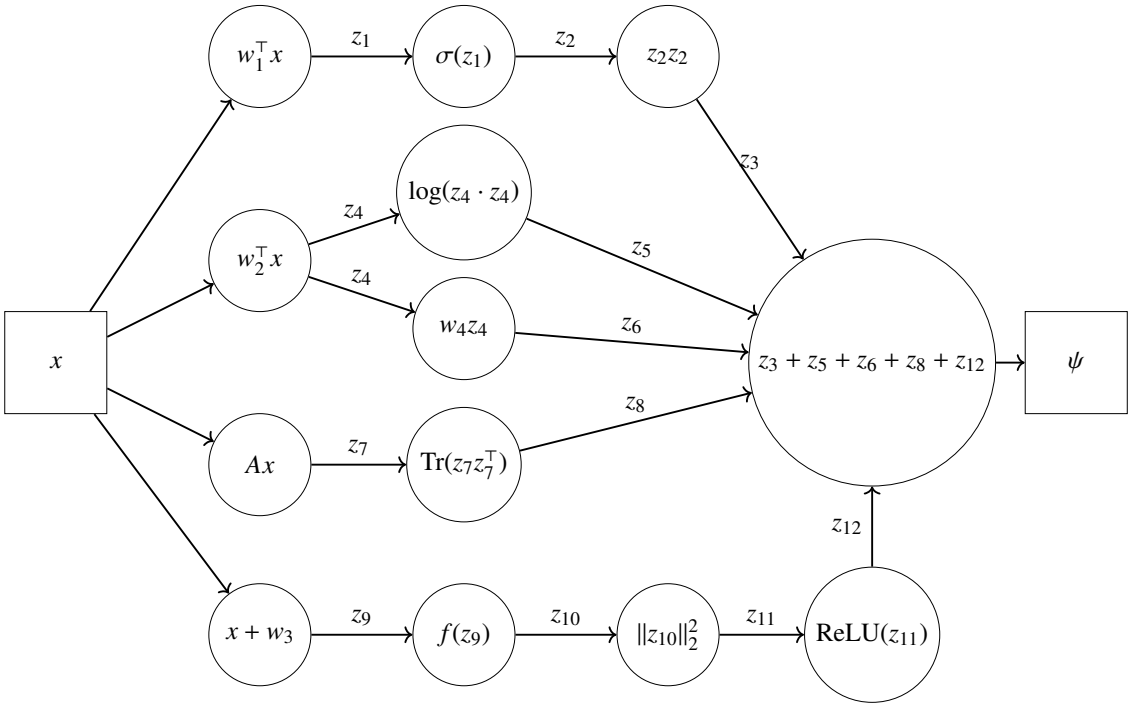
$$\begin{aligned} p(\theta \mid x) &= \frac{p(x \mid \theta)p(\theta)}{p(x)} \\ &\propto p(x \mid \theta)p(\theta) \\ &= \frac{1}{B(\alpha,\beta)}(1-\theta)^{x-1}\theta\theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &\propto \theta^{(\alpha+1)-1}(1-\theta)^{(\beta+x-1)-1} \end{aligned}$$

Comparing the posterior to the functional form of the Beta distribution PDF, we see that $\gamma = \alpha + 1$ and $\delta = \beta + x - 1$. The posterior is Beta($\alpha + 1, \beta + x - 1$).

initial here

4 Neural Networks and Backpropagation

Abby is experimenting with a nonstandard neural network architecture, depicted below.



The input $x \in \mathbb{R}^d$ and output $\psi \in \mathbb{R}$ are enclosed in rectangles. The parameters are:

$$w_1, w_2, w_3 \in \mathbb{R}^d \qquad w_4 \in \mathbb{R} \qquad A \in \mathbb{R}^{d \times d}$$

Computations are enclosed in circles. Intermediate activations are denoted as z_i and drawn on their corresponding edges. In this figure, $\sigma : \mathbb{R} \to \mathbb{R}$ is the sigmoid activation function and $f : \mathbb{R}^d \to \mathbb{R}^d$ denotes a function that subtracts the mean of a vector from itself:

$$\sigma(y) = \frac{1}{1 + e^{-y}} \qquad f(y) = y - \frac{1}{d} \sum_{i=1}^{d} (y)_i \mathbf{1}_d$$

in which $\mathbf{1}_d$ is the all-ones vector in $\mathbb{R}^d$ and $(y)_i$ denotes the i^{th} entry of y.

Abby now wants to train her network. In this problem, you will compute some of the derivatives needed to do so. Write all answers in terms of variables present in Figure 1.

(a) (2 points) $\frac{\partial \psi}{\partial w_1}$

Solution:

$$\begin{aligned} \frac{\partial \psi}{\partial w_1} &= \frac{\partial \psi}{\partial z_3} \frac{\partial z_3}{\partial z_2} \frac{\partial z_2}{\partial z_1} \frac{\partial z 1}{\partial w_1} \\ &= 2 z_2 \sigma(z_1)(1 - \sigma(z_1)) x^\top \end{aligned}$$

initial here

(b) (2 points) $\frac{\partial \psi}{\partial w_2}$

Solution:

$$\begin{aligned}\frac{\partial \psi}{\partial w_2} &= \frac{\partial \psi}{\partial z_5}\frac{\partial z_5}{\partial z_4}\frac{\partial z_4}{\partial w_2} + \frac{\partial \psi}{\partial z_6}\frac{\partial z_6}{\partial z_4}\frac{\partial z_4}{\partial w_2} \\ &= \frac{2z_4}{z_4^2}x^\top + w_4 x^\top \\ &= (\frac{2z_4}{z_4^2} + w_4)x^\top\end{aligned}$$

(c) (3 points) $\frac{\partial \psi}{\partial A}$

Solution: We write the derivative as a function of a matrix Y.

$$\begin{aligned}\frac{\partial \psi}{\partial A}(Y) &= \frac{\partial \psi}{\partial z_8}\frac{\partial z_8}{\partial z_7}\frac{\partial z_7}{\partial A}(Y) \\ &= 2z_7^\top(Y)x \\ &= \mathrm{Tr}(2z_7^\top(Y)x) \\ &= \mathrm{Tr}(2xz_7^\top(Y))\end{aligned}$$

Hence, the derivative is $2xz_7^\top$.

(d) (3 points) $\frac{\partial \psi}{\partial w_3}$

Solution: Note that the z_{11} is always nonnegative, so $z_{12} = z_{11}$.

$$\begin{aligned}\frac{\partial \psi}{\partial z_{12}}\frac{\partial z_{12}}{\partial z_{11}}\frac{\partial z_{11}}{\partial z_{10}}\frac{\partial z_{10}}{\partial z_9}\frac{\partial z_9}{\partial w_3} &= 2z_{10}^\top\left(I_d - \frac{1}{d}11^\top\right)I \\ &= 2z_{10}^\top\left(I_d - \frac{1}{d}11^\top\right)\end{aligned}$$

initial here

5 Monitoring Regression

Kermit has found a bin of unused computer monitors lying around the EECS department. Many of the monitors are broken, but a few are still functional. Kermit decides to build a simple model to predict whether a given monitor will work. He encodes monitor i as a single feature $x_i \in \{-1, 1\}$ depending on its color. He also records whether monitor i is functional with a single indicator label $y_i \in \{0, 1\}$. Specifically,

$$x_i = \begin{cases} 1 & \text{if monitor } i \text{ is beige} \\ -1 & \text{otherwise} \end{cases} \qquad y_i = \begin{cases} 1 & \text{if monitor } i \text{ is functional} \\ 0 & \text{otherwise} \end{cases}$$

Kermit checks N monitors total and aggregates his findings into a dataset $X \in \{-1, 1\}^N$ with labels $Y \in \{0, 1\}^N$:

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^\top \qquad Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}^\top$$

(a) (2 points) Let $n_{x,y}$ denote the number of monitors for which $x_i = x$ and $y_i = y$. For example, $n_{+1,0}$ denotes the number of non-functional beige monitors. Specifically:

$$n_{-1,0} = \left|\{i \mid x_i = -1 \text{ and } y_i = 0\}\right| \qquad n_{-1,1} = \left|\{i \mid x_i = -1 \text{ and } y_i = 1\}\right|$$
$$n_{+1,0} = \left|\{i \mid x_i = 1 \text{ and } y_i = 0\}\right| \qquad n_{+1,1} = \left|\{i \mid x_i = 1 \text{ and } y_i = 1\}\right|$$

Show that the solution to the linear regression objective

$$w_{\text{OLS}} = \text{argmin}_{w \in \mathbb{R}} \|Xw - Y\|_2^2$$

is given by

$$w_{\text{OLS}} = \frac{n_{+1,1} - n_{-1,1}}{n}.$$

Solution: The solution to linear regression is given by

$$(X^\top X)^{-1} X^\top Y = \left(\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \right)^{-1} \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$
$$= (n)^{-1} \sum_{i=1}^{n} x_i \mathbf{1}\left[y_i = 1\right]$$
$$= \frac{n_{1,1} - n_{-1,1}}{n}$$

initial here

(b) (3 points) Recall the logistic regression problem in one dimension:

$$\text{argmin}_{w\in\mathbb{R}} f(w) = \text{argmin}_{w\in\mathbb{R}} \left[-\sum_{i=1}^{n} y_i \log\left(\sigma(wx_i)\right) + (1-y_i)\log\left(1-\sigma(wx_i)\right) \right]$$

in which σ is the sigmoid function, $\sigma(z) = \frac{1}{1+e^{-z}}$. Show that

$$\frac{df(w)}{dw} = X^\top(\sigma(Xw) - Y)$$

in which $\sigma(Xw)$ denotes the result of applying σ elementwise to Xw.

Solution:

$$\begin{aligned}
\frac{\partial f(w)}{\partial w} &= -\sum_{i=1}^{n} \frac{y_i\sigma(wx_i)(1-\sigma(wx_i))x_i}{\sigma(wx_i)} - \frac{(1-y_i)\sigma(wx_i)(1-\sigma(wx_i)x_i}{1-\sigma(wx_i)} \\
&= -\sum_{i=1}^{n} y_i(1-\sigma(wx_i)x_i - (1-y_i)\sigma(wx_i)x_i \\
&= -\sum_{i=1}^{n} y_i x_i - \sigma(wx_i)x_i \\
&= \sum_{i=1}^{n} x_i(\sigma(wx_i) - y_i) \\
&= X^\top(\sigma(Xw) - Y)
\end{aligned}$$

initial here

(c) (4 points) Let

$$\hat{p} = \frac{n_{+1,1} + n_{-1,0}}{n}.$$

Show that a solution to the logistic regression problem in the previous part is given by:

$$w = \log\left(\frac{\hat{p}}{1-\hat{p}}\right).$$

Solution: When $x_i = 1$,

$$\begin{aligned}
\sigma(wx_i) &= \sigma(w) \\
&= \sigma\left(\log\left(\frac{\hat{p}}{1-\hat{p}}\right)\right) \\
&= \frac{1}{1 + e^{-\log\left(\frac{\hat{p}}{1-\hat{p}}\right)}} \\
&= \frac{1}{1 + \frac{1-\hat{p}}{\hat{p}}} \\
&= \hat{p}.
\end{aligned}$$

Similarly, when $x_i = -1$,

$$\begin{aligned}
\sigma(wx_i) &= \sigma(-w) \\
&= \sigma\left(\log\left(\frac{1-\hat{p}}{\hat{p}}\right)\right) \\
&= 1 - \hat{p}.
\end{aligned}$$

Hence,

$$\begin{aligned}
X^\top \sigma(Xw) &= \sum_{i=1}^{n} \mathbf{1}\left[x_i = 1\right]\hat{p} - \mathbf{1}\left[x_i = -1\right](1-\hat{p}) \\
&= (n_{1,0} + n_{1,1})\hat{p} - (n_{-1,0} + n_{-1,1})(1-\hat{p}) \\
&= (n_{1,0} + n_{1,1})\hat{p} + (n - n_{1,0} - n_{1,1})(\hat{p} - 1) \\
&= n_{1,0}\hat{p} + n_{1,1}\hat{p} + n\hat{p} - n - n_{1,0}\hat{p} + n_{1,0} - n_{1,1}\hat{p} + n_{1,1} \\
&= n\hat{p} - n + n_{1,0} + n_{1,1} \\
&= n_{1,1} + n_{-1,0} - n + n_{1,0} + n_{1,1} \\
&= n_{1,1} - n_{-1,1}
\end{aligned}$$

Recalling from part (a) that $X^\top Y = n_{1,1} - n_{-1,1}$, we see that $X^\top \sigma(Xw) = X^\top Y$, so the derivative $\frac{\partial f(w)}{\partial w}$ is 0, and thus w is a solution to the logistic regression problem.

initial here

6 Classification with hedging

Consider a classification problem with c classes. The input is denoted as x and the label as $y \in \{1, 2, \ldots, c\}$. Suppose we allow our prediction to take one of $2c$ values:

$$1, 2, \ldots, c, 1_{\text{hedge}}, 2_{\text{hedge}}, \ldots, c_{\text{hedge}}.$$

The i_{hedge} option represents a not-entirely-sure guess for class i. When one hedges one's bets by predicting i_{hedge} instead of i, the reward for being correct decreases, but the penalty for being wrong also decreases.

$$L(f(x), y) = \begin{cases} 1 & \text{if } f(x) \in \{1, 2, \ldots, c\} \text{ and } f(x) \neq y \\ 0 & \text{if } f(x) \in \{1, 2, \ldots, c\} \text{ and } f(x) = y \\ \lambda_e & \text{if } f(x) \in \{1_{\text{hedge}}, 2_{\text{hedge}}, \ldots, c_{\text{hedge}}\} \text{ and } f(x) \neq y \\ \lambda_f & \text{if } f(x) \in \{1_{\text{hedge}}, 2_{\text{hedge}}, \ldots, c_{\text{hedge}}\} \text{ and } f(x) = y \end{cases}$$

where $0 < \lambda_e, \lambda_f < 1$. The risk of a classifier f evaluated at input x is

$$R(f(x) \mid x) = \sum_{i=1}^{c} L(f(x), i)\, P(Y = i \mid x).$$

(a) (1 point) Let us assume that $\lambda_f < \lambda_e$. Explain why this is a reasonable assumption.

Solution: This is saying that the loss for guessing correctly is less than the loss for guessing wrong; if we want to encourage correct guesses, of course λ_f should be less than λ_e.

initial here

(b) (2 points) Find the minimizers

$$\operatorname*{argmin}_{i\in\{1,2,\ldots,c\}} R(i \mid x) \quad \text{and} \quad \operatorname*{argmin}_{i\in\{1,2,\ldots,c\}} R(i_{\text{hedge}} \mid x).$$

Express your answers as simple statements involving the conditional label probabilities $P(Y = i \mid x)$ for $i \in \{1, 2, \ldots, c\}$.

Solution:

$$\begin{aligned} R(i \mid x) &= \sum_{j=1}^{c} L(i,j) P(Y = j \mid x) \\ &= \sum_{j=1}^{c} 1[i \neq j] P(Y = j|x) \\ &= 1 - P(Y = i|x) \\ R(i_{\text{hedge}} \mid x) &= \lambda_e \sum_{j\neq i} P(Y = j|x) + \lambda_f P(Y = i|x) \\ &= \lambda_e + P(Y = i|x)(\lambda_f - \lambda_e). \end{aligned}$$

In both cases, the coefficient of $P(Y = i \mid x)$ is negative, so minimizing the risk is the same as maximizing $P(Y = i \mid x)$. Thus, the argmin is at the maximal probability class:

$$i^* = \operatorname*{argmax}_{i\in\{1,2,\ldots,c\}} P(Y = i \mid x)$$

initial here

(c) (2 points) Given some $i \in \{1, 2, \ldots, c\}$, when is $R(i_{\text{hedge}} \mid x) \leq R(i \mid x)$? Express your answer as an inequality in which the left hand side is $P(Y = i \mid x)$ and the right hand side is an expression involving λ_e and λ_f.

Solution: Writing out the inequality, we have

$$\begin{aligned}
&R(i_{\text{hedge}} \mid x) \leq R(i \mid x) \\
\iff &\lambda_e P(Y \neq i \mid x) + \lambda_f P(Y = i \mid x) \leq P(Y \neq i \mid x) \\
\iff &\lambda_e (1 - P(Y = i \mid x)) + \lambda_f P(Y = i \mid x) \leq 1 - P(Y = i \mid x) \\
\iff &\lambda_e + P(Y = i \mid x)(\lambda_f - \lambda_e) \leq 1 - P(Y = i \mid x) \\
\iff &P(Y = i \mid x)(1 - \lambda_e + \lambda_f) \leq 1 - \lambda_e \\
\iff &P(Y = i \mid x) \leq \frac{1 - \lambda_e}{1 - \lambda_e + \lambda_f}.
\end{aligned}$$

(d) (1 point) Give an intuitive explanation for what happens when $\lambda_f = 0$.

Solution: When $\lambda_f = 0$, $R(i_{\text{hedge}} \mid x) \leq R(i \mid x)$ when $P(Y = i \mid x) \leq 1$. In other words, the risk from hedging (predicting i_{hedge}) is always less than or equal to the risk from not hedging (predicting class i). This is the case because the loss from a correct prediction is the same (0), but the penalty from an incorrect prediction is lower when hedging. Therefore, one should always hedge.

(e) (2 points) Find a risk-minimizing predictor $f^*(x)$.

Solution: Once again, let

$$i^* = \underset{i \in \{1,2,\ldots,c\}}{\text{argmax}} P(Y = i \mid x).$$

In part (b), we also established that if we do not hedge, we should predict class i^*, and that if we do hedge, we should predict ${i^*}_{\text{hedge}}$. The condition for choosing between hedging and not hedging is given in part (c).

$$f^*(x) = \begin{cases} {i^*}_{\text{hedge}} & \text{iff } P(Y = i \mid x) \leq \frac{1-\lambda_e}{1-\lambda_e+\lambda_f} \\ i^* & \text{otherwise} \end{cases}$$

initial here

7 Visualizing high-dimensional data (CS289A only)

Only complete this problem if you are enrolled in CS289A.
Do **not** complete this problem if you are enrolled in CS189.

In this problem, we will explore an approach for visualizing N high-dimensional datapoints $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, $\mathbf{x}_i \in \mathbb{R}^d$ as two-dimensional embeddings $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$, $\mathbf{y}_i \in \mathbb{R}^2$.

(a) (2 points) Given a point $\mathbf{x}_i$, we would like to model a distribution over all other points $\mathbf{x}_j$ ($i \neq j$). The probability of sampling point $\mathbf{x}_j$ should decrease as $\mathbf{x}_j$ gets further away from $\mathbf{x}_i$, so we begin by defining a Gaussian distribution centered at each of the points $\mathbf{x}_i$:

$$\mathbb{P}(\mathbf{x} \mid \mathbf{x}_i) = \mathcal{N}(\mathbf{x} \mid \mathbf{x}_i, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^\top \Sigma^{-1}(\mathbf{x} - \mathbf{x}_i)\right\}.$$

Let $\Sigma = \sigma^2 I_d$ for all such distributions. The probability of sampling point $\mathbf{x}_j$ given point $\mathbf{x}_i$ is then given by normalizing the probabilities under the Gaussian data-generating process:

$$\mathrm{p}_{ij} = \frac{\mathbb{P}(\mathbf{x}_j \mid \mathbf{x}_i)}{\sum_{k,k\neq i} \mathbb{P}(\mathbf{x}_k \mid \mathbf{x}_i)}.$$

Show that p_{ij} can be expressed as,

$$\mathrm{p}_{ij} = \frac{\exp\left(-d_{ij}^2\right)}{\sum_{k,k\neq i} \exp\left(-d_{ik}^2\right)}$$

in which

$$d_{ij}^2 = \frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}.$$

Solution: Simply plugging and chugging the expression suffices.

$$\begin{aligned}
\mathrm{p}_{ij} &= \frac{\mathbb{P}(\mathbf{x}_j \mid \mathbf{x}_i)}{\sum_{k\neq i} \mathbb{P}(\mathbf{x}_k \mid \mathbf{x}_i)} \\
&= \frac{\frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left\{d_{ij}^2\right\}}{\sum_{k\neq i} \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left\{d_{ik}^2\right\}} \\
&= \frac{\exp\left(-d_{ij}^2\right)}{\sum_{k\neq i} \exp\left(-d_{ik}^2\right)}
\end{aligned}$$

initial here

We now define an analogous conditional sampling distribution over the two-dimensional points $\mathbf{y}_j$. We will assume a covariance of $\Sigma = \frac{1}{2}I$ for the two-dimensional Gaussian distributions to simplify notation. Specifically, let the probability of sampling point $\mathbf{y}_j$ given point $\mathbf{y}_i$ be given by:

$$q_{ij} = \frac{E_{ij}}{Z_i}$$

in which

$$E_{ij} := \exp\left(-\hat{d}_{ij}^2\right) \qquad Z_i := \sum_{\substack{k \\ k \neq i}} \exp\left(-\hat{d}_{ik}^2\right) \qquad \hat{d}_{ij}^2 = \left\|\mathbf{y}_i - \mathbf{y}_j\right\|^2$$

E and Z are often referred to as the energy and partition functions in machine learning.

(b) (2 points) The Kullback-Leibler (KL) divergence, $D_{\mathrm{KL}}(P \parallel Q)$, is a popular statistical distance to measure the difference between two distributions P and Q. KL divergence is given by

$$D_{\mathrm{KL}}(\mathrm{p}_i \parallel \mathrm{q}_i) := \sum_j \mathrm{p}_{ij} \log \frac{\mathrm{p}_{ij}}{\mathrm{q}_{ij}}$$

Show that KL divergence penalizes a mismatch of a large p_{ij} with a small q_{ij} more severely than a mismatch of small p_{ij} with a large q_{ij}. For simplicity and in this part only, consider only a single term in the sum without trying to reason over the entire sum. Explain in a few words why this is a desirable property of a loss function that aims to model the local structure in data.

Solution: Let $0 < c_2 < c_1 < 1$. Suppose for some, j, $\mathrm{p}_{ij} = c_1$ and $\mathrm{q}_{ij} = c_2$. The contribution of this mismatch to the cost is $c_1 \log \frac{c_1}{c_2} > 0$, as $\frac{c_1}{c_2} > 1$. In the reverse case of, $\mathrm{p}_{ij} = c_2$ and $\mathrm{q}_{ij} = c_1$, contribution to the cost is $c_2 \log \frac{c_2}{c_1} < 0$, as $\frac{c2}{c1} < 1$ and $c_1, c_2 > 0$. Hence, $c_1 \log \frac{c_1}{c_2} > 0 > c_2 \log \frac{c_2}{c_1}$.

This is a desirable property of the loss function because this biases optimization of q_{ij} to favor matching high p_{ij} more closely than low p_{ij}, and a high p_{ij} represents x_j being a close neighbour of x_i. Hence, this causes optimization to model the local structure in x better, at the cost of the global structure.

initial here

Define the cost of overall mapping to be the sum of KL divergences over the data,

$$C = \sum_i D_{\mathrm{KL}}\left(\mathrm{p}_i \| \mathrm{q}_i\right) = \sum_i \sum_{\substack{j \\ j \neq i}} \mathrm{p}_{ij} \log \mathrm{p}_{ij} - \sum_i \sum_{\substack{j \\ j \neq i}} \mathrm{p}_{ij} \log \mathrm{E}_{ij} + \sum_i \sum_{\substack{j \\ j \neq i}} \mathrm{p}_{ij} \log \mathrm{Z}_i$$

We would like to optimize this cost function with respect to the two-dimensional embeddings $\mathbf{y}_k$ (everything else is fixed) with a gradient based optimization procedure like SGD.

(c) (0.5 points) First show that,

$$\frac{\partial \log E_{kj}}{\partial \mathbf{y}_k} = 2(\mathbf{y}_j - \mathbf{y}_k)$$

Solution: Denote $\frac{\partial}{\partial \mathbf{y}_k}$ as ∂ for brevity.

$$\partial \log E_{kj} = \frac{\partial E_{kj}}{E_{kj}} = \frac{E_{kj}\, \partial(-\hat{d}_{kj}^2)}{E_{kj}} = \frac{2E_{kj}(\mathbf{y}_j - \mathbf{y}_k)}{E_{kj}} = 2(\mathbf{y}_j - \mathbf{y}_k)$$

(d) (0.5 points) Now show that,

$$\frac{\partial Z_i}{\partial \mathbf{y}_k} = \begin{cases} \frac{\partial E_{ik}}{\partial \mathbf{y}_k} & \text{if } i \neq k \\ \sum_{j, j \neq k} \frac{\partial E_{kj}}{\partial \mathbf{y}_k} & \text{if } i = k \end{cases}$$

Solution: Once again denote $\frac{\partial}{\partial \mathbf{y}_k}$ as ∂ for brevity.

$$\partial Z_i = \partial \sum_{j \neq i} E_{ij} = \begin{cases} \partial \sum_{j \neq i} E_{ij} & i \neq k \\ \partial \sum_{j \neq i} E_{kj} & i = k \end{cases} = \begin{cases} \partial E_{ik} & i \neq k \\ \sum_{j \neq k} \partial E_{kj} & i = k \end{cases}$$

initial here

(e) (3.0 points) Finally show that,

$$\frac{\partial C}{\partial \mathbf{y}_k} = 2\sum_j \left(\mathbf{y}_k - \mathbf{y}_j\right)\left(p_{kj} - q_{kj} + p_{jk} - q_{jk}\right)$$

Solution:

$$\partial C = -\partial \sum_i \sum_{j,i\neq j} \mathrm{p}_{ij} \log \mathrm{E}_{ij} + \partial \sum_i (\sum_{j,i\neq j} \mathrm{p}_{ij}) (\log \mathrm{Z}_i) \tag{3}$$

$$= -\sum_{j,j\neq k} \underbrace{\partial \left(\mathrm{p}_{kj} \log \mathrm{E}_{kj} + \mathrm{p}_{jk} \log \mathrm{E}_{jk}\right)}_{\text{Term 1}} + \underbrace{\sum_i \partial \log Z_i}_{\text{Term 2}} \qquad \left[\sum_{j,j\neq i} \mathrm{p}_{ij} = 1\right] \tag{4}$$

Using part (c) with the first term,

$$\partial \left(\mathrm{p}_{kj} \log \mathrm{E}_{kj} + \mathrm{p}_{jk} \log \mathrm{E}_{jk}\right) = \mathrm{p}_{kj} \partial \left(\log \mathrm{E}_{kj}\right) + \mathrm{p}_{jk} \partial \left(\log \mathrm{E}_{jk}\right) = 2(p_{kj} + \mathrm{p}_{jk})(\mathbf{y}_j - \mathbf{y}_k)$$

Considering the second term with part (d),

$$\sum_i \partial \log Z_i = \sum_j \partial \log Z_j = \sum_j \frac{\partial Z_j}{Z_j} = \left(\sum_{j,j\neq k} \frac{\partial Z_j}{Z_j}\right) + \frac{\partial Z_k}{Z_k} = 2\sum_{j,j\neq k} q_{jk}(\mathbf{y}_j - \mathbf{y}_k) + 2\sum_{j,j\neq k} q_{kj}(\mathbf{y}_j - \mathbf{y}_k)$$

Putting it together,

$$\partial C = \sum_{j,j\neq k} 2\left(\mathbf{y}_k - \mathbf{y}_j\right)\left(p_{kj} - q_{kj} + p_{jk} - q_{jk}\right) \tag{5}$$

$$= \sum_j 2\left(\mathbf{y}_k - \mathbf{y}_j\right)\left(p_{kj} - q_{kj} + p_{jk} - q_{jk}\right) \tag{6}$$

gives the required gradient.

(f) (2.0 points) Find the Big $O(\cdot)$ time complexity of an efficient implementation of one step of gradient descent on the data embeddings $\{\mathbf{y}_i\}_{i=1}^N$. Write the complexity in terms of the number of datapoints N and their dimensionality d.

Solution: The calculation of the gradient first involves calculating all of the N^2 interactions between all possible (i, j) pairs. Then, all the row-wise sums (and equivalently column-wise sums) are calculated from this distance matrix in $O(N^2)$ time. For each $\mathbf{y}_k$, $\sum_j q_{kj}$ can be calculated by simply subtracting the correct $\exp\left(-d_{kj}^2\right)$ from the row sum to obtain the denominator of each q_{kj}. This allows calculating $\sum_j q_{kj}$ in $O(N)$ time. Hence, each $\frac{\partial C}{\partial \mathbf{y}_k}$ can be calculated in $O(N)$ time.

Thus, the gradient for all data points can be calculated in $O(N^2 d)$ time for computing pairwise distance and $O(N^2)$ time for caching row sums of the pairwise distance matrix, and finally, $O(N^2)$ time for computing the gradient itself from the caches. Hence, the overall time complexity is $O(N^2 d)$.

This page intentionally left blank.

This page intentionally left blank.

This page intentionally left blank.

This page intentionally left blank.