CS 189/289A Introduction to Machine Learning
Fall 2024 Jennifer Listgarten, Saeed Saremi

Final

- 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 two 8.5 × 11 inch cheat sheets.
- You have 2 hours 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 2 hours 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. If you run out of space, you may use the last four pages to continue showing your work.
- **The last question is for CS289A students only**. Students enrolled in CS189 will **not** receive any credit for answering this question.

Your Name	
Your SID	
Name and SID of student to your left	
Name and SID 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. Which of the following statements about logistic regression is **false**?

   ○ Logistic regression models the log-odds as a linear function of the features.

   ○ The output of a logistic regression model is always between 0 and 1.

   ○ The "steep" part of the sigmoid function (inputs around -0.1 to 0.1) can cause vanishing gradients.

   ○ Logistic regression can be used for classification.

   Solution: The correct answer is C.

   Vanishing gradients occur at the "flat" parts of the sigmoid function. A is true by definition. B is true due to the nature of the sigmoid function. D is true as the outputs of logistic regression can be interpreted as probabilities, so we can predict the class with the highest probability.

2. Which of the following statements about Decision Trees is **true**?

   ○ The predictions of two Decision Trees in a Random Forest are completely independent of each other.

   ○ A single Decision Tree in a Random Forest always overfits on the random subset of the data it is trained on.

   ○ A common Decision Tree training objective is minimizing the misclassification rate.

   ○ Decision Trees do not benefit from input normalization, as they are invariant to the scale of the features.

   Solution: The correct answer is D. Decision Trees make splits based on finding the best threshold value; the scale of the value does not matter.

   A is wrong because the randomization makes the *errors*, not the Decision Trees themselves, uncorrelated. B is wrong because, in a Random Forest, Decision Trees also train on a random subset of the features, which counteracts overfitting. C is wrong because most Decision Trees are trained to minimize the *entropy*, not the misclassification rate. Many splits could have the same misclassification rate, which could be better distinguished via entropy.

3. Which of the following statements about k-means clustering is **true**?

   ○ k-means requires fewer parameters than Mixture of Gaussians.

   ○ k-means labels new points using the majority label of the k closest points.

   ○ k-means can accurately identify clusters of arbitrary shapes and densities.

   ○ k-means assigns each data point to a cluster by minimizing the pairwise distances between all data points.

   Solution: The correct answer is A. Beyond the cluster means (centroids) parameterized in k-means, Mixture of Gaussians additionally requires tracking the cluster covariances and cluster weights.

B is incorrect because it describes k nearest neighbors, not k-means. Recall that k-means is an unsupervised algorithm that does not have access to labels. C is incorrect because k-means assumes clusters are spherical in shape which causes it to struggle with clusters of arbitrary shapes or densities. D is incorrect because each data point is assigned to the cluster of the nearest centroid, not by minimizing pairwise distances among all points.

4. Bob is running Langevin MCMC to sample from a distribution $p(x) \sim N(\mu, \Sigma)$ using its score function, where $x \in \mathbb{R}^2$. To improve the algorithm's convergence, he hyperparameter tunes the step size $\eta = [\eta_x, \eta_y]^T$ such that $\eta_x > \eta_y$. What is most likely to be **true** regarding the shape of the target distribution $p(x)$?

   ○ The mean $\mu = [1, 2]^T$ and covariance $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

   ○ The mean $\mu = [2, 1]^T$ and covariance $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

   ○ The mean $\mu = [0, 0]^T$ and covariance $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

   ○ The mean $\mu = [0, 0]^T$ and covariance $\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$

   Solution: The correct answer is D.

   Tuning the step size such that $\eta_x > \eta_y$ helps the most if the target distribution is anistropic and elongated along the x-axis. A and B are wrong because their covariances Σ imply that $p(x)$ is isotropic, or spherical. C is wrong because its covariance Σ implies that $p(x)$ is an ellipse elongated along the y-axis.

5. Two binary classifiers, f_1 and f_2, are trained on the same dataset but evaluated on separate datasets. f_1 is evaluated on dataset $\mathcal{D}_1$, where 30% of the instances are from the positive class. f_2 is evaluated on dataset $\mathcal{D}_2$, where 70% of the instances are from the positive class. Both models achieve an Area Under the Receiver Operating Characteristic (AUROC) of 0.8 on their respective datasets. Which of the following statements is **true**?

   ○ Compared to f_1, f_2 has a higher probability of correctly ranking a randomly chosen positive instance above a randomly chosen negative instance.

   ○ Both classifiers have identical true positive rates and false positive rates at all classification thresholds.

   ○ The AUROC of the two classifiers cannot be compared because the two evaluation datasets have different proportions of positive instances.

   ○ Both classifiers have the same capability to discriminate between positive and negative examples.

   Solution: The correct answer is D.

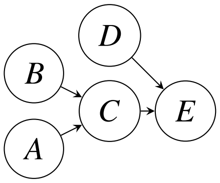
   A. False. AUROC measures the probability that a classifier will assign a higher score to a randomly chosen positive instance than to a randomly chosen negative one. Both classifiers have an AUROC of 0.8. Therefore f_2 does not have a higher probability than f_1.

B. False. There are multiple ROC curves with the same area underneath them. Therefore, the two classifiers are not guaranteed to have identical TPR and FPR values at all thresholds.

C. False. AUROC is insensitive to the class distribution of the evaluation datasets. Therefore, the differing proportions of positive instances in $\mathcal{D}_1$ and $\mathcal{D}_2$ do not impede the comparison of the AUROCs. Note that other factors, such as the inherent difficulty of the datasets, might influence the meaningfulness of comparing AUROCs, but these factors are unrelated to class proportions.

D. True. Both classifiers are equally proficient in ranking positive instances higher than negative ones.

6. Consider the graphical model displayed below



Which of the following independence statements **does not always hold true**?

○ A and B are marginally independent.

○ B and E are conditionally independent given C.

○ A and D are marginally independent.

○ A and B are conditionally independent given C.

Solution: The correct answer is D.

A. There is only one path that goes from A to B, $A \to C \leftarrow B$, which is blocked because it is a v-structure. A and B are marginally independent.

B. There is one path that goes from B to E, which is blocked by C.

C. The path $A \to C \to E \leftarrow D$ is blocked at E because it is a v-structure; A and D are marginally independent.

D. Because C is in the conditioning set, the path $A \to C \leftarrow B$ is not blocked, because it is a v-structure.

7. Consider a Hidden Markov Model with hidden states $X_1, \dots, X_T$ and corresponding observations $Y_1, \dots, Y_T$ for each time step t in 1 to T. Assume there are N possible hidden states.

The Viterbi algorithm uses dynamic programming to determine the most probable sequence of hidden states given the observations. Which of the following statements about the Viterbi algorithm is **false**?

○ The Viterbi algorithm initializes the dynamic programming table using the initial state distribution $P(X_1)$ and the emission probability $P(Y_1 \mid X_1)$.

○ The most probable hidden state at time t depends only on the observations from Y_1 to Y_t and not the observations from Y_{t+1} to Y_T.

○ The Viterbi algorithm maintains backpointers at each time step to facilitate the reconstruction of the most probable state sequence after processing all observations.

○ The Viterbi algorithm has a computational complexity of $O(N^2 \cdot T)$, where N is the number of hidden states and T is the length of the observation sequence.

Solution: The correct answer is B. While the Viterbi algorithm processes observations sequentially from time 1 to T, determining the most probably hidden state at time t requires determining how well it aligns with future observations Y_{t+1} to Y_T. This is why we need to do backtracing from time T as opposed to choosing the hidden state at each time point with the largest Viterbi path probability.

8. Which of the following statements about the discounting factor γ in reinforcement learning is **true**?

○ A discount rate $\gamma > 1$ can be used to emphasize future rewards more than immediate rewards.

○ Lowering the discount rate can encourage an agent to prioritize short-term rewards over long-term rewards.

○ The optimal discount rate can be chosen via Maximum Likelihood Estimation.

○ When computing returns, the discount rate only applies to terminal rewards (attained by landing in a terminal state) and not any intermediate rewards.

Solution: The correct answer is B.

Option A is wrong because the discount rate is bounded in the range $[0, 1]$. Option C is wrong because the discount rate is a hyperparameter that must be chosen beforehand. Option D is wrong because the discounting factor applies to all rewards except the immediate reward when computing the returns, regardless of whether the environment is finite or infinite horizon.

9. Which of the following statements about optimal policies for finite MDPs is **false**?

○ All optimal policies achieve the same state-value function $v(s)$.

○ All optimal policies achieve the same action-value function $q(s, a)$.

○ If all rewards are doubled, the set of optimal policies changes.

○ There always exists an optimal policy for a finite MDP that is completely deterministic (a deterministic policy picks exactly one action in each state).

Solution: The correct answer is C.

Option A is true because that is how an optimal policy is defined: any policy that achieves the optimal state-value function $v_*(s) = \max_\pi v_\pi(s)$ is considered an optimal policy. Option B is true because the optimal action-value function can be related to the optimal state-value function via $q_*(s,a) = \sum_{r,s'} p(r, s' \mid s, a)(r + \gamma v_*(s'))$. Option C is false because doubling all rewards will double the expected returns, i.e., the value function, and $\text{argmax}_\pi v_\pi(s) = \text{argmax}_\pi 2v_\pi(s)$. Option D is true because one can just run value iteration until it converges to get a completely deterministic optimal policy.

10. Which of the following statements is **false** regarding kernel methods:

○ Kernel methods can only be applied to supervised learning tasks.

○ Kernel methods implicitly embed data points in potentially infinite-dimensional spaces.

○ Kernel methods can be applied to non-conventional input spaces, such as graphs or strings.

○ Kernel methods scale at least quadratically with the number of training samples.

Solution: The correct answer is A.

A. False. Kernel methods can be applied to both supervised and unsupervised learning tasks, e.g., kernel PCA.

B. True. The representer theorem shows that kernels can be represented as inner products of embeddings, which can be potentially infinite-dimensional.

C. True. Kernel methods are flexible and can be applied to various types of data, including graphs and strings.

D. True. Kernel methods require computing the kernel matrix K where $K_{ij} = k(x_i, x_j)$. Computing this matrix requires at least $O(n^2)$ time where n is the number of training samples.

Consider the ridge regression objective $J(w) = \|Xw - y\|_2^2 + \lambda\|w\|_2^2$ for some $\lambda > 0$. Let w^* denote the minimizer of the above expression. Which of the following is **true**?

11. ○ $Xw^* = y$.

○ w^* exists if and only if X^TX is invertible.

○ $w^* = X^\dagger y$, where $X^\dagger$ is the pseudo-inverse of X.

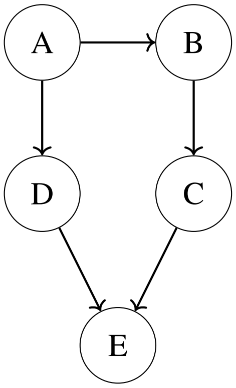
○ The minimizer w^* is unique.

Solution: The correct answer is D.

Note that $w^* = (X^TX + \lambda I)^{-1}X^Ty$, which does not require that X^TX is invertible. It also does not imply that $Xw^* = y$ or that $w^* = X^\dagger y$.

However, the minimizer w^* is unique. The Hessian of $J(w)$ is $2X^TX + 2\lambda I$, which is positive definite as X^TX is positive semi-definite and λI is positive definite.

12. Which of the following is **true** of the following causal graph (graph corresponding to an SEM)?



○ E is a mediator between D and C

○ D and C are confounded by E.

○ A is confounded by E.

○ B and D are confounded by A.

Solution: The correct answer is D.

Two variables are confounded if they have a common parent in a causal graph. Hence, choice D is correct, and choices B and C are incorrect. A variable Y is a mediator between variables X and Z if the paths $X \rightarrow Y \rightarrow Z$ or $X \leftarrow Y \leftarrow Z$ appear in the graph. Hence, choice A is incorrect (E is a *collider* between D and C).

13. Which of the following is **false** about graphical neural networks (GNNs)?

○ Choice of AGGREGATE and COMBINE functions determine the GNN.

○ A valid AGGREGATE function applied with a normalization based on the number of neighbors is still permutation invariant.

○ The node-level predictions of a GNN are permutation equivariant.

○ GNNs can be trained to distinguish non-isomorphic graphs that the Weisfeiler-Lehman (WL) test cannot.

Solution:

The correct answer is D.

Option D is false since GNNs cannot exceed the performance of the WL test. Options A and C are fundamental properties of GNNs. Option B is true since changing the ordering of vertices does not change the degree (number of neighbors) of a vertex.

14. You are training a convolutional neural network with five layers to classify different plant species. The model is initialized with random weights. Which of the following is most likely to show the strongest clustering of images by their class when t-SNE is applied?

○ The output of the **first** layer of the network **prior** to training.

○ The output of the **first** layer of the network **after** the model has been trained.

○ The output of the **fourth** layer of the network **prior** to training.

○ The output of the **fourth** layer of the network **after** the model has been trained.

Solution: The correct answer is D.

Prior to training, the neural network acts randomly on the input data, so we do not expect any strong clustering after applying t-SNE. Hence, A and C are incorrect. After training, we expect the neural network to develop more "confident" representations of the data as it goes through the network, and we expect these clusters to be strongest near the end of the network.

Students have seen this behavior in the t-SNE demo in homework 4.

15. In convolutional neural networks, max pooling is used to downsample feature maps. However, max pooling can also introduce challenges during training. Which of the following drawbacks of max pooling is **true**?

○ Max pooling increases the number of model parameters, thereby making the model more prone to overfitting.

○ Max pooling can disproportionately emphasize noisy activations, potentially leading to unstable feature representations.

○ Max pooling changes the activation functions of neurons from linear to non-linear, complicating the training process.

○ Max pooling is a non-differentiable operation, preventing the use of gradient-based methods for optimization.

Solution: The correct answer is B.

A. False, Max pooling layers do not introduce any additional learnable parameters.

B. True. Unlike average pooling, which smooths out activations by taking their mean, max pooling selects the highest activation within a window. This can cause the model to focus on outlier or noisy activations, potentially resulting in less stable and less reliable feature representations.

C. False. Max pooling does not alter the activation functions. It is typically applied after activation functions.

D. False. Max pooling is differentiable. During backpropagation, gradients are propagated only to the input neuron that had the maximum value in the pooling window, while gradients for other neurons in the window are set to zero.

16. Which of the following is **true** of regularization methods?

○ Ridge regression induces sparser solutions than LASSO.

○ Transforming data by using *all* the principal components of X instead of X itself is a form of regularization.

○ Regularization aims to reduce both the bias^2 and variance of a model.

○ Adding Gaussian noise to data can be interpreted as a form of regularization.

Solution: The correct answer is D.

Students have seen that noising data yields a shrinkage effect in homework 7. Choice A is reversed–LASSO yields sparser solutions. B is false as the full PC matrix is full-rank and orthonormal, so multiplying XV simply rotates or reflects the data. C is false since regularization increases bias.

Note: During the exam, we clarified that X is full rank.

initial here

2 Short Answer

1. (3 points) Suppose we have n i.i.d. samples $X_1, X_2, \dots, X_n$ drawn from the same distribution $X \sim \text{Poisson}(\theta)$. Our goal is to estimate the mean θ with the estimator $\hat{X}$. We define $\hat{X}$ to be:

$$\hat{X} = \frac{\alpha + \sum_{i=1}^{n} X_i}{\beta + n}$$

where α and β are constants greater than 0. What is the bias and variance of the estimator $\hat{X}$? *Hint:* The mean and variance of the Poisson distribution are the same, i.e., $\mathbb{E}[X] = \text{Var}[X] = \theta$.

(a) (1.5 points) What is the **bias** of the estimator $\hat{X}$?

Hint: The expectation of a Poisson(θ) random variable is θ.

Solution: The bias can be written as follows

$$\begin{aligned}\mathbb{E}\left[\hat{X} - \theta\right] &= \mathbb{E}\left[\frac{\alpha + \sum_{i=1}^{n} X_i}{\beta + n} - \theta\right] \\ &= \frac{\alpha + n \cdot \theta}{\beta + n} - \theta \\ &= \frac{\alpha - \beta \cdot \theta}{\beta + n} \\ &= \frac{\alpha}{\beta + n} - \frac{\beta}{\beta + n} \cdot \theta\end{aligned}$$

Note: During the exam, we clarified that answers should only depend on α, β, n, and θ.

(b) (1.5 points) What is the **variance** of the estimator $\hat{X}$?

Hint: The variance of a Poisson(θ) random variable is θ.

Solution: The variance can be written as follows

$$\begin{aligned}\text{Var}\left[\hat{X}\right] &= \text{Var}\left[\frac{\alpha + \sum_{i=1}^{n} X_i}{\beta + n}\right] \\ &= \frac{1}{(\beta + n)^2} \text{Var}\left[\sum_{i=1}^{n} X_i\right] \\ &= \frac{1}{(\beta + n)^2} \sum_{i=1}^{n} \text{Var}\left[X_i\right] \\ &= \frac{n}{(\beta + n)^2} \cdot \theta\end{aligned}$$

where we used the fact that, if random variables A, B are independent, then $\text{Var}[A + B] = \text{Var}[A] + \text{Var}[B]$.

Note: During the exam, we clarified that answers should only depend on α, β, n, and θ.

2. (2 points) Suppose you have three data points in $\mathbb{R}$:

$$\{x_1 = 0, x_2 = 6, x_3 = 12\}$$

initial here

Suppose you run k-means clustering with $k = 2$ and at a particular iteration observe the following clustering:

$$C_1 = \{x_2\}, C_2 = \{x_1, x_3\}$$

What are the centroids, and what is the total sum of squared errors? If this is the current cluster assignment, does the k-means algorithm terminate or keep going?

Solution: We are given the clusters $C_1 = \{x_2\}, C_2 = \{x_1, x_3\}$.

The centroids are $c_1 = 6, c_2 = \frac{0+12}{2} = 6$.

The sum of squared errors is $\text{SSE} = (6-6)^2 + (0-6)^2 + (12-6)^2 = 36 + 36 = 72$.

Because both centroids are $c_1 = c_2 = 6$, there is no change in assignments and the k-means algorithm terminates. However, this is a local (not global) optimum as there exist other clusterings with a lower sum of squared errors. For example, $C_1 = \{x_1\}, C_2 = \{x_2, x_3\}$ yields an SSE of 18.

3. (2 points) Suppose a convolutional layer has the following specifications:

- Input dimensions: [height = 20, width = 20, channels = 3]
- Output dimensions: [height = 10, width = 10, channels = 10]
- Kernel size: [height = 5, width = 5]
- Each kernel also contains an additional bias term.

How many trainable parameters are there in this convolutional layer?

Solution: The correct answer is 760.

Each kernel has $(5 \times 5 \times 3 + 1) = 76$ parameters, where the additional one represents the bias term. There are 10 such kernels, so in total there are $76 \times 10 = 760$ parameters.

4. (2 points) Suppose that we have a dataset of points in $\mathbb{R}$ where each point comes from either class A or class B, both of which are one-dimensional Gaussians with equal prior probabilities:

- Class A: $\mu_A = 0, \sigma_A^2 = 1$
- Class B: $\mu_B = 2, \sigma_B^2 = 4$.

Determine the decision boundary $x \in \mathbb{R}$ where a point is equally likely to belong to class A or class B. You do not need to solve for x exactly. Instead express it as the solution of a quadratic equation $ax^2 + bx + c = 0$ for some $a, b, c \in \mathbb{R}$.

Solution:

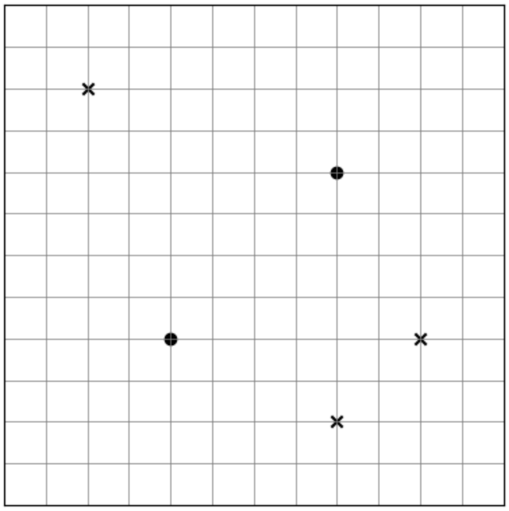
$$\begin{aligned}
p(c_A \mid x) &= p(c_B \mid x) \\
\frac{p(x \mid c_A)p(c_A)}{p(x)} &= \frac{p(x \mid c_B)p(c_B)}{p(x)} \\
p(x \mid c_A) &= p(x \mid c_B) \\
\frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}x^2\right] &= \frac{1}{2\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-2}{2}\right)^2\right]
\end{aligned}$$

initial here

$$2\exp\left[-\frac{1}{2}x^2\right] = \exp\left[-\frac{1}{2}\left(\frac{x-2}{2}\right)^2\right]$$
$$\ln 2 - \frac{1}{2}x^2 = -\frac{1}{2}\left[\frac{x^2}{4} - x + 1\right]$$
$$-\frac{3}{8}x^2 - \frac{1}{2}x + \ln 2 + \frac{1}{2} = 0$$

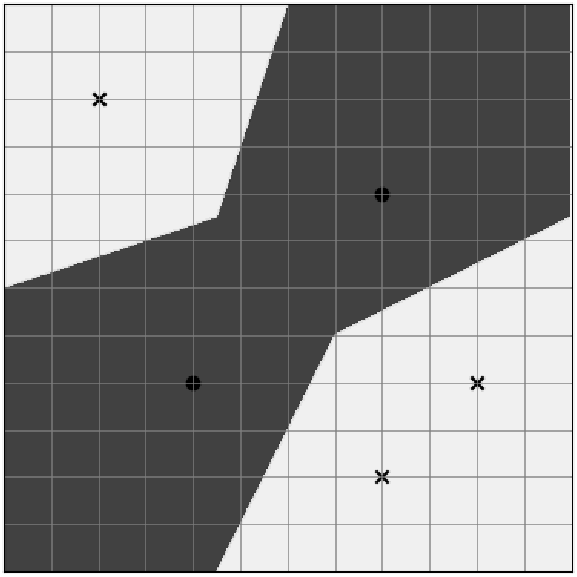
Therefore, we have that $a = -\frac{3}{8}$, $b = -\frac{1}{2}$ and $c = \ln 2 + \frac{1}{2}$. Any other values a', b', c' that are multiples of a, b, c would also yield the correct answer.

5. (2 points) Consider the following training dataset of points from two classes: dots and crosses. Draw the decision boundary that will be returned by a 1 Nearest Neighbor classifier. Assume that the distance metric being used here is regular Euclidean (ℓ^2) distance.



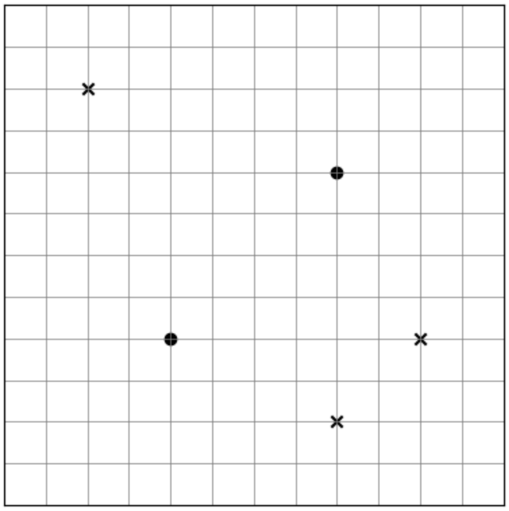
Solution: Here is what the decision boundary will look like:

initial here

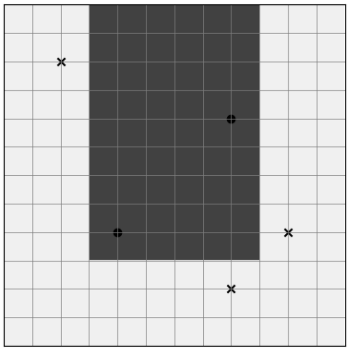


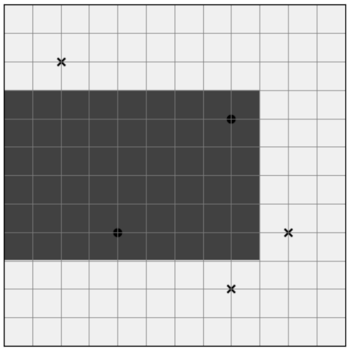
While this plot shows the only correct boundary that can achieved by a 1NN classifier, we will award full credit to students who are able to get the general shape correct, even if not all lines pass through or intersect at the exact grid points expected. Since we are using ℓ^2 distance, the boundary between two points of opposing classes will be their perpendicular bisector.

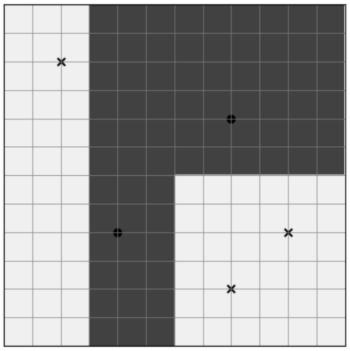
6. (2 points) Consider the same training dataset as above. Draw a plausible decision boundary that will be returned by a Decision Tree classifier with no bound on its max depth. You may assume that trees are trained using the same greedy procedure shown in lecture, that you also implemented in homework 5.



Solution: There are many correct answers; as long as both classes are separated by 3 axis-aligned line segments, it will be a valid decision boundary. Here are some possible boundaries:







Decision boundaries that are simply a box around the two points that belong to the dotted class will not be considered valid: these boundaries will require at least 4 splits, but the greedy algorithm for training a decision tree will terminate in at most 3 splits.

7. (2 points) Consider a dataset of two features f_1 and f_2 consisting of the six sample points

$$\begin{bmatrix} 4 & 6 & 9 & 1 & 7 & 5 \\ 1 & 6 & 5 & 2 & 3 & 4 \end{bmatrix}^\top$$

with labels

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}^\top.$$

We build a decision tree of depth 1 by choosing the single split that maximizes the information gain.

Define j as the feature and v as the value to split on at the root node (so if a value is $\geq v$, then we put it in the right tree, and if it is $< v$, then we put it in the left tree). For the optimal first (and only) split, what is the corresponding j and v?

Solution: Let's look at the features first and see what the optimal splits per feature might look like. If we sort by f_1, the features and the corresponding labels are

$$\begin{bmatrix} 1 & 4 & 5 & 6 & 7 & 9 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

If we sort by f_2, we have

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

By inspection, $f_1 \geq 7$ is the best split since it separates 0s and 1s the best. Therefore, the correct answer is $j = 1$ and $v = 7$.

We can also explicitly calculate the information gain, which is defined as

$$H(Y) - H(Y \mid X_{j,v}).$$

The entropy before the split is equal to

$$H(Y) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1.$$

The conditional entropy after the split is equal to

$$\begin{aligned}
H(Y \mid X_{1,7}) &= P(X_{1,7} = 1)H(Y \mid X_{1,7} = 1) + P(X_{1,7} = 0)H(Y \mid X_{1,7} = 0) \\
&= \frac{1}{3}\left[-1 \log_2(1)\right] + \frac{2}{3}\left[-\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right] \\
&= 0 - \frac{1}{2}\log_2\left(\frac{3}{4}\right) - \frac{1}{6}(-2) \\
&= -\frac{1}{2}\log_2\left(\frac{3}{4}\right) + \frac{1}{3}
\end{aligned}$$

Therefore, the information gain is $\frac{2}{3} + \frac{1}{2}\log_2\left(\frac{3}{4}\right)$.

8. (3 points) Consider an MDP with 3 states C (center), L (left) and R (right), and two actions l (left) and r (right). Here are the transition dynamics described qualitatively:

   - In state C, picking the left action l will transition you to the left state L, and will receive a reward of $+1$.
   - In state C, picking the right action r will transition you to the right state R, and will receive a reward of 0.
   - In state L, picking either action will transition you back to state C, and will receive a reward of 0.
   - In state R, picking either action will transition you back to state C, and will receive a reward of $+2$.

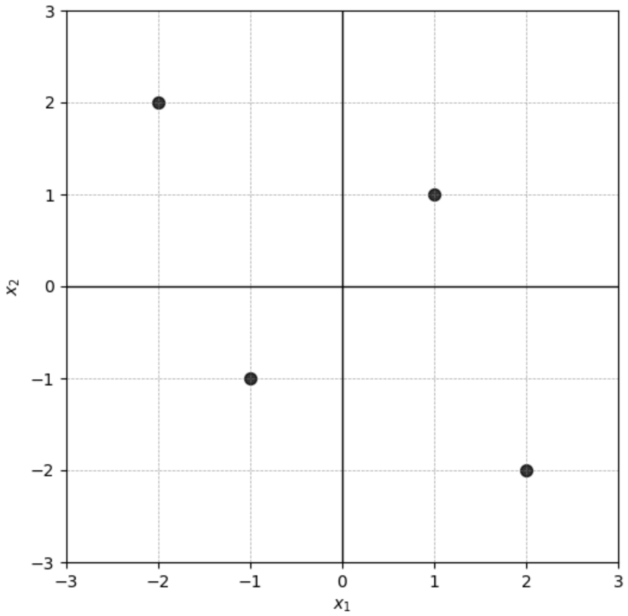
   Describe an optimal policy for this MDP when the discount rate is: (a) $\gamma = 0.0$, (b) $\gamma = 0.9$ and (c) $\gamma = 0.5$. You can reason about the policies intuitively (in fact, it might be helpful to draw this MDP) and you should not have to compute them algorithmically. Assume that an episode always starts in state C.

   Solution: When $\gamma = 0.0$, we maximize immediate rewards so the optimal policy will choose action l in state C and receive a reward of $+1$. Once in state L, it can choose either action since they both transition back to state C anyways. It will never land in state R, so we can choose either action for this state.

   When $\gamma = 0.9$, we maximize long-horizon rewards so the optimal policy will choose action r in state C to land at node R since any action taken from there will yield a reward of $+2$. In state R, it can take any action since they both transition back to state C anyways. It will never land in state L, so we can choose either action for this state.

   When $\gamma = 0.5$, both policies above will be equally optimal and achieve the same returns, and we can consider some mixture of them (for example, alternating between action pairs (l, any) and (r, any)).

9. (2 points) Consider the following four points in $\mathbb{R}^2$.



(a) (1 point) We perform PCA on the data. What is one possible first principal component?

Solution: Most variance is captured in the direction of the line $x_2 = -x_1$, so either of

$$\pm\frac{1}{\sqrt{2}}\begin{bmatrix}-1\\1\end{bmatrix}$$

is a valid answer.

(b) (1 point) Suppose we wish to reconstruct the point

$$\begin{bmatrix}-1\\2\end{bmatrix}$$

using the first principal component. That is, we want to compute the vector projection of our point onto the first PC. What is the resulting vector? (Your answer should be in $\mathbb{R}^2$.)

Solution: Let x denote our point $\begin{bmatrix}-1\\2\end{bmatrix}$ and v denote the principal component found in part (a). Then, the projection of x onto v is

$$\begin{aligned}(x^T v)v &= \frac{3}{\sqrt{2}}\begin{bmatrix}-\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{bmatrix}\\ &= \begin{bmatrix}-1.5\\1.5\end{bmatrix}\end{aligned}$$

10. (3 points) Consider a hidden Markov model of the weather, with states $x_t \in \{\text{sunny}, \text{rainy}\}$ and observations $o_t \in \{\text{dry}, \text{wet}, \text{drenched}\}$, with the prior, transition, and emission probabilities:

x_1	$P(x_1)$
sunny	0.5
rainy	0.5

x_t	x_{t+1}	$P(x_{t+1}\|x_t)$
sunny	sunny	0.8
sunny	rainy	0.2
rainy	sunny	0.4
rainy	rainy	0.6

x_t	o_t	$P(o_t\|x_t)$
sunny	dry	0.9
sunny	wet	0.1
rainy	dry	0.2
rainy	wet	0.5
rainy	drenched	0.3

(a) (2 points) What is the probability of the sequence of observations (drenched, drenched, dry)? You may leave your answer as a product of numbers.

Solution: For shorthand, let R and S denote rainy and sunny states, and $D1$, W, $D2$ denote dry, wet, and drenched observations. The observation drenched can only occur when it is rainy. This implies the first two states must be rainy. Thus,

$$\begin{aligned}
P(x_{1,2,3} = (R, R, R)) &= P(x_1 = R)P(x_2 = R|x_1 = R)P(x_3 = R|x_2 = R) \\
&= 0.5 \cdot 0.6 \cdot 0.6 = 0.18 \\
P(x_{1,2,3} = (R, R, S)) &= P(x_1 = R)P(x_2 = R|x_1 = R)P(x_3 = S|x_2 = R) \\
&= 0.5 \cdot 0.6 \cdot 0.4 = 0.12 \\
P(o_{1,2,3}|x_{1,2,3} = (R, R, R)) &= P(o_1 = D2|x_1 = R)P(o_2 = D2|x_2 = R)P(o_3 = D1|x_3 = R) \\
&= 0.3 \cdot 0.3 \cdot 0.2 = 0.018 \\
P(o_{1,2,3}|x_{1,2,3} = (R, R, S)) &= P(o_1 = D2|x_1 = R)P(o_2 = D2|x_2 = R)P(o_3 = D1|x_3 = S) \\
&= 0.3 \cdot 0.3 \cdot 0.9 = 0.081 \\
P(o_{1,2,3} = (D2, D2, D1)) &= P(o_{1,2,3}|x_{1,2,3} = (R, R, R))P(x_{1,2,3} = (R, R, R)) \\
&\quad + P(o_{1,2,3}|x_{1,2,3} = (R, R, S))P(x_{1,2,3} = (R, R, S)) \\
&= 0.18 \cdot 0.018 + 0.12 \cdot 0.081 = 0.01296.
\end{aligned}$$

Note: During the exam, we clarified that students could leave their answer as a "sum of product of numbers," as opposed to the original wording which simplify said "product of numbers."

(b) (1 point) What is the most likely sequence of states given the sequence of observations (drenched, drenched, dry)?

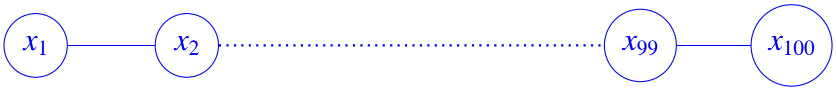
Solution: Based on the calculations above, (rainy, rainy, sunny) is the most likely sequence.

11. (2 points) Suppose you are training a graph neural network (GNN) to predict whether a molecule is soluble or not. You represent each molecule as a graph where each atom is a node, and edges exist between nodes if there is a chemical bond between them. Each molecule in your dataset contains at most 100 atoms, and there always exists a path between any pair of nodes (i.e. the graph is connected).

(a) (1 point) Determine the maximum number of GNN layers required so that information is shared between all pairs of nodes (even if there is not a direct edge between them) for any molecule in your dataset.

Solution: The worst case scenario involves a chain graph.

initial here



Suppose we need to send information from x_1 to x_{100}. After layer 1, x_2 will have information about x_1. After layer 2, x_3 will have information about x_1 via x_2. Continuing this, after layer 99, x_{100} will finally get information about x_1. Therefore, the maximum number of layers needed is 99.

(b) (1 point) **Note: this part is independent of the previous part.** In part (a), you found that your GNN requires a large number of layers to allow information to be shared between all pairs of nodes. If you constructed a GNN with this many layers and used a sigmoid activation function, what potential issue could arise during training? Name one strategy to prevent this without adding new edges between nodes.

Solution: Because of the large number of layers and the use of a sigmoid activation function, we might run into vanishing gradients. To avoid this, we can use residual (skip) connections between the same node at different layers or use a ReLU activation function instead.

We will also give full credit to those who said that the large number of layers could cause exploding gradients. Again, using residual (skip) connections would be a valid recourse. Other strategies like gradient clipping or weight decay would also be accepted. Changing the activation function to ReLU (or the like) is not a valid strategy.

initial here

3 MLE and MAP of the exponential distribution (6 points)

The probability density function of an exponential distribution is

$$f_{\text{exponential}}(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Here, $\lambda > 0$ is the parameter of the distribution.

(a) (2 points) Suppose that we observe n independent and identically distributed examples, denoted as $(x_1, x_2, \ldots, x_n)$, from an exponential distribution with parameter λ, where $x_i \geq 0$ for all i. Derive the log-likelihood function for the parameter λ given these observations.

Solution:

$$\mathcal{L}(\lambda) = \prod_{i=1}^{n} \lambda e^{-\lambda x_i}$$

$$\begin{aligned} \log \mathcal{L}(\lambda) &= \sum_{i=1}^{n} \log \left[\lambda e^{-\lambda x_i}\right] \\ &= \sum_{i=1}^{n} \log \lambda - \lambda x_i \\ &= n \log \lambda - \sum_{i=1}^{n} \lambda x_i \end{aligned}$$

(b) (2 points) Compute the maximum likelihood estimate of λ. You can assume without proof that the log likelihood is concave.

Solution:

$$\begin{aligned} \frac{\partial}{\partial \lambda} \log \mathcal{L}(\lambda^{\text{MLE}}) &= 0 \\ \frac{n}{\lambda^{\text{MLE}}} - \sum_{i=1}^{n} x_i &= 0 \\ \frac{n}{\lambda^{\text{MLE}}} &= \sum_{i=1}^{n} x_i \\ \lambda^{\text{MLE}} &= \frac{n}{\sum_{i=1}^{n} x_i} \end{aligned}$$

(c) (2 points) Suppose we have a prior on λ that it comes from a Gamma(α,β) distribution for given values of α and β. The probability density function of a Gamma distribution is

$$f_{\text{gamma}}(x;\alpha,\beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}.$$

initial here

Show that the posterior distribution $p(\lambda \mid x_1, \ldots, x_n)$ follows a Gamma$(n+\alpha, \beta + \sum_{i=1}^n x_i)$ distribution.

Hint: You may show that the posterior is proportional to the density of the desired Gamma distribution.

Solution:

$$\begin{aligned}
p(\lambda \mid x_1, \ldots, x_n) &\propto L(\lambda)p(\lambda) \\
&= \left(\prod_{i=1}^n \lambda e^{-\lambda x_i}\right)\left(\frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}\right) \\
&= \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{n+\alpha-1}e^{-(\beta+\sum_{i=1}^n x_i)\lambda}
\end{aligned}$$

Therefore, the posterior is equivalent to a Gamma$(n+\alpha, \beta + \sum_{i=1}^n x_i)$ distribution.

4 Re-Learning Self-Attention (7 points)

OpenCorp is a technology company with a proprietary model that consists of a single self-attention layer with a single head. The company provides an API that returns the intermediate attention weights generated by the model for any given input. However, the model's original weight matrices remain undisclosed, and your goal is to re-learn the key and query weight matrices, W_k and W_q, solely by calling the API.

We will initialize the matrices W_k and W_q randomly and re-learn them using gradient ascent. We call the API n times to create a dataset, $D = \{X^{(i)}, A^{(i)}\}_{i=1}^{n}$, where i denotes the sample index. Each input $X^{(i)}$ is a sequence of L tokens, represented as a matrix whose jth row corresponds to the jth token. Each $A^{(i)} \in \mathbb{R}^{L \times L}$ is a matrix containing the corresponding ground-truth attention weights.

To train our weight matrices, we characterize the self-attention layer forward pass as follows

$$\begin{aligned} K^{(i)} &= X^{(i)} W_k^\top \\ Q^{(i)} &= X^{(i)} W_q^\top \\ \hat{A}^{(i)} &= \text{Softmax}(Q^{(i)} K^{(i)\top}) \end{aligned}$$

where $\hat{A}^{(i)} \in \mathbb{R}^{L \times L}$ is the matrix of predicted attention weights computed using our current estimates for W_k and W_q, and the softmax is applied row-wise. We will interpret the attention weights $\hat{A}^{(i)}$ and $A^{(i)}$ as probabilities because they normalize to 1.

(a) (1 point) A naive way to learn W_k and W_q is by minimizing squared error between $A^{(i)}$ and $\hat{A}^{(i)}$, which are to be interpreted as probabilities. Briefly explain why minimizing mean squared error (MSE) might be a poor choice for an objective function in this scenario.

Solution: MSE is a poor choice for learning probabilities. To name a few:

- Probabilities represent distributions, and MSE treats the output as numerical values without accounting for their probabilistic nature (probabilities must be non-negative and sum to 1).
- MSE operates on a linear scale, and penalizes differences in probabilities equally. For example, it will penalize a difference of 0.9 vs 0.8 equally as a difference of 0.0 vs 0.1, even though the impact on the overall distribution alignment may be very different.
- MSE leads to smaller gradients when the predicted probability is closer to the target. This can slow learning for confident predictions that are slightly incorrect, making it harder for models to refine their outputs.

Alternative objectives like cross-entropy better capture the probabilistic nature of attention weights.

(b) (2 points) Write down a better objective function $\mathcal{L}(W_k, W_q)$ that we can maximize (rather than minimize) to recover W_k and W_q from D.
Hint: once again, attention weights $A_{pq}^{(i)}$ and $\hat{A}_{pq}^{(i)}$ can be interpreted as probabilities.

Solution: Note that the attention weights $\hat{A}^{(i)}$ and $A^{(i)}$ are all probabilities. Thus, we can minimize the KL-Divergence between them. However, this is equivalent to maximizing the (negative) cross-entropy between them, i.e., we can let our objective function be

$$\mathcal{L}(W_k, W_q) = \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{L}\sum_{l=1}^{L} A_{tl}^{(i)} \log \hat{A}_{tl}^{(i)}$$

Maximizing the objective above will adjust the weight matrices W_k and W_q until the computed attention weights $\hat{A}^{(i)}$ align with the provided attention weights $A^{(i)}$. Any variant of the objective above will be deemed acceptable.

(c) (2 points) How would you modify your objective function if we want to explicitly encourage the queries for each token in a sequence to be orthogonal to each other?
Hint: formulate and add a penalty term to your objective from part (b). Denote λ to be the hyperparameter associated with the weight of this penalty term.

Solution: Following the notation above, let $q_t^{(i)}$ denote the t-th row of $Q^{(i)}$, i.e., the query corresponding to the t-th token of sequence i. Two query vectors being orthogonal is equivalent to the condition $(q_t^{(i)})^\top(q_l^{(i)}) = 0$. Therefore, in order to encourage orthogonality, we would like to drive the magnitude of all such pairwise inner products to 0. This can be achieved by adding a penalty term to the objective above, of the form

$$-\lambda \sum_{i=1}^{n}\sum_{t\neq l}((q_t^{(i)})^\top(q_l^{(i)}))^2$$

where $\lambda > 0$ is some hyperparameter that we can control. There is a negative sign at the front of the penalty because we are maximizing the objective in part (a), and each term within the summation above is positive. Our new objective function is, thus,

$$\mathcal{L}(W_k, W_q) = \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{L}\sum_{l=1}^{L} A_{tl}^{(i)} \log \hat{A}_{tl}^{(i)} - \lambda \sum_{i=1}^{n}\sum_{t\neq l}((q_t^{(i)})^\top(q_l^{(i)}))^2$$

We will also accept an L1 version of the penalty, i.e., replacing each $((q_t^{(i)})^\top(q_l^{(i)}))^2$ with $|(q_t^{(i)})^\top(q_l^{(i)})|$. The penalty term is what's being graded in this subpart and, to avoid double jeopardy, a reasonably correct penalty will earn full credit regardless of the objective from part (b).

(d) (2 points) **Note: this subpart is independent of the previous subparts.**
After closer inspection, we realize that the learned attention weights are not causally masked such that $\hat{A}_{tl}^{(i)} = 0$ for $l > t$. Your friend from Stanford points out that one way to enforce causal masking is to multiply $\hat{A}^{(i)}$ with a mask matrix M, where $M_{tl} = 1$ for $l \leq t$ and 0 otherwise. Is this a valid strategy? If not, suggest one fix to make this masking strategy valid.

Solution: This is not a valid strategy. If we simply multiply our attention weights with a masking matrix, the resulting masked attention weights will not contain valid probabilities that row-wise sum up to 1 anymore. One fix would be to re-normalize the probabilities after masking so they row-wise sum up to 1. Another would be to apply the mask to pre-softmax unnormalized attention scores by setting the masked entries to a large negative value, so they get zeroed out post-softmax. Turns out that both of these methods are mathematically equivalent.

5 Naive Bayes classification (8 points)

The naive Bayes model is a generative model used for classification. We let X denote observed features, and X_i denote the i-th feature (out of d features in total). We let $C \in \{1, 2, \ldots, K\}$ denote the class label. The classifier can be represented using the following directed acyclic graph:

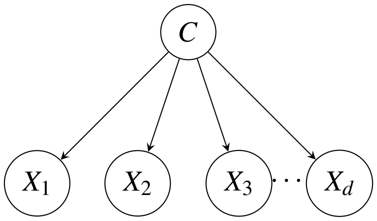


Figure 1: Naive Bayes Graph

(a) (1 point) Write the joint distribution $P(X_1, X_2, \ldots, X_d, C)$ of the graph in simplest form.

Solution: Based on the graph, the joint distribution of Naive Bayes is given by

$$P(X_1, X_2, \ldots, X_d, C) = P(C) \prod_{j=1}^{d} P(X_j \mid C).$$

(b) (1.5 points) Let i, j denote two distinct feature indices. Are X_i and X_j necessarily conditionally independent given C? You may assume that all the features are discrete in this question. Correct answers with incomplete reasoning will not receive full credit.

Solution: X_i and X_j are conditionally independent given C. To see why, we can marginalize out all features X_k for $k \neq i, j$ from the joint distribution:

$$\begin{aligned} P(X_i, X_j, C) &= \sum_{X_{k \neq i,j}} P(C) \prod_{k=1}^{d} P(X_k \mid C) \\ &= P(C) P(X_i \mid C) P(X_j \mid C). \end{aligned}$$

This implies that

$$\begin{aligned} P(X_i, X_j \mid C) &= \frac{P(X_i, X_j, C)}{P(C)} \\ &= P(X_i \mid C) P(X_j \mid C). \end{aligned}$$

This shows that X_i and X_j are conditionally independent given C.

(c) (1.5 points) Are X_i and X_j necessarily independent? You can again assume that all the features are discrete in this question.

Solution: However, X_i and X_j are not marginally independent. To see why, we note that

$$P(X_i, X_j) = \sum_{C} P(X_i, X_j, C)$$

$$= \sum_C P(C)P(X_i \mid C)P(X_j \mid C),$$

which has no reason to be equal to $P(X_i)P(X_j)$ without additional assumptions.

For a precise example, consider the variables $X_i = X_j = C$, with C non-degenerate. Then clearly, X_i and X_j are not independent as X_i has the same distribution as C, but given $X_j = a$, we have $X_i = a$ with probability 1, so $P(X_i \mid X_j) \neq P(X_i)$

Note: During the exam, we clarified that students must provide a mathematical justification for their answer as they are required to in part (b).

(d) (3 points) Assume now that $P(C = k) = \frac{1}{K}$ for all k. We also assume that conditional on $C = k$, each X_i is i.i.d. from a Bernoulli distribution with parameter $\theta_k \in (0, 1)$ that is already known. Our goal is to classify a new observation based on its features, $X_1, X_2, \ldots, X_d$.

Express $P(C = k \mid X_1, X_2, \ldots, X_d)$ in terms of $\theta_1, \ldots, \theta_K$ and $X_1, \ldots, X_d$.

Solution: We have that

$$P(C = k \mid X_1, X_2, \ldots, X_d) = \frac{P(X_1, X_2, \ldots, X_d, C = k)}{P(X_1, X_2, \ldots, X_d)}.$$

The numerator is given by

$$P(X_1, X_2, \ldots, X_d, C = k) = \frac{1}{K} \prod_{j=1}^{d} \theta_k^{X_j}(1 - \theta_k)^{1-X_j}.$$

Using the law of total probability, the denominator is given by

$$P(X_1, X_2, \ldots, X_d) = \sum_{k'=1}^{K} P(X_1, X_2, \ldots, X_d, C = k') = \sum_{k'=1}^{K} \left[\frac{1}{K} \prod_{j=1}^{d} \theta_{k'}^{X_j}(1 - \theta_{k'})^{1-X_j} \right],$$

which gives the final expression

$$P(C = k \mid X_1, X_2, \ldots, X_d) = \frac{\prod_{j=1}^{d} \theta_k^{X_j}(1 - \theta_k)^{1-X_j}}{\sum_{k'=1}^{K} \prod_{j=1}^{d} \theta_{k'}^{X_j}(1 - \theta_{k'})^{1-X_j}}.$$

(e) (1 point) How would you classify a new observation based on its features? You can either write an optimization problem or explain your strategy (a single sentence should suffice).

Solution:

One strategy to classify a data point with features $X_1, X_2, \ldots, X_d$ consists in predicting the class with maximal posterior probability:

$$\hat{C} = \operatorname*{argmax}_k \; P(C = k \mid X_1, X_2, \ldots, X_d).$$

initial here

6 Langevin MCMC for Logistic Regression (11 points)

Consider a logistic regression model, for $x,\ \beta \in \mathbb{R}^d$ and $y \in \{-1, 1\}$, where the likelihood is

$$p(y \mid x, \beta) = \sigma(y \cdot \beta^\top x)$$

where $\sigma(\cdot)$ represents the sigmoid function and the prior on the weights is Gaussian

$$\beta \sim N(0, \sigma^2 I)$$

In this question, we will explore why sampling from the posterior $p(\beta \mid x, y)$ is challenging. We will consider how Langevin Markov Chain Monte Carlo (MCMC) sampling offers a potential solution, using the score function $\nabla_\beta \log p(\beta \mid x, y)$.

(a) (2 points) Show that the posterior can be written as follows:

$$p(\beta \mid x, y) = \frac{p(y \mid x, \beta) p(\beta)}{p(y \mid x)}$$

Assume that β, x are independent.

Solution:

$$\begin{aligned} p(\beta \mid x, y) &= \frac{p(\beta, x, y)}{p(x, y)} \\ &= \frac{p(y \mid x, \beta) p(\beta \mid x) p(x)}{p(y \mid x) p(x)} \\ &= \frac{p(y \mid x, \beta) p(\beta)}{p(y \mid x)} \end{aligned}$$

(b) (2 points) Write the normalization term $p(y \mid x)$ in terms of $p(y \mid x, \beta)$ and $p(\beta)$. Explain why computing this normalization constant can be computationally intractable.

Solution: To compute the normalization constant, we have to integrate over all possible values of β

$$p(y \mid x) = \int p(y \mid x, \beta)\, p(\beta)\, d\beta$$

In many cases, there is no closed form expression for the integral. It is often computationally intractable to integrate over all values of β, especially if β is high dimensional, due to the curse of dimensionality.

(c) (3 points) Compute the following gradients. (These will be useful to us in the following parts.)

i. $\nabla_\beta \log p(y \mid x, \beta)$

ii. $\nabla_\beta \log p(\beta)$

iii. $\nabla_\beta \log p(y \mid x)$

Solution:

(i) We know that $p(y \mid x, \beta) = \sigma(y \cdot \beta^\top x)$.

$$\begin{aligned}\nabla_\beta \log \sigma(y \cdot \beta^\top x) &= \frac{\nabla_\beta \sigma(y \cdot \beta^\top x)}{\sigma(y \cdot \beta^\top x)} \\ &= y \cdot x \cdot \frac{\sigma(y \cdot \beta^\top x)(1 - \sigma(y \cdot \beta^\top x))}{\sigma(y \cdot \beta^\top x)} \\ &= y \cdot x \cdot \left(1 - \sigma(y \cdot \beta^\top x)\right)\end{aligned}$$

(ii) We know that $\beta \sim N(0, \sigma^2 I)$.

$$\begin{aligned}\nabla_\beta \log p(\beta) &= \nabla_\beta \log \left[\frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|\beta - 0\|^2}{2\sigma^2}\right)\right] \\ &= \nabla_\beta \left[-\frac{\|\beta - 0\|^2}{2\sigma^2}\right] \\ &= -\frac{2\beta}{2\sigma^2} \\ &= -\frac{\beta}{\sigma^2}\end{aligned}$$

(iii) We know that $p(y \mid x)$ is not a function of β.

$$\nabla_\beta \log p(y \mid x) = 0$$

(d) (2 points) Using the previous part, write an expression for the score function $\nabla_\beta \log p(\beta \mid x, y)$. Why might the score function be easier to compute than the posterior?

Solution: We can rewrite the score function of the posterior as follows:

$$\begin{aligned}\nabla_\beta \log p(\beta \mid x, y) &= \nabla_\beta \log p(y \mid x, \beta) + \nabla_\beta \log p(\beta) - \nabla_\beta \log p(y \mid x) \\ &= y \cdot x \cdot \left(1 - \sigma(y \cdot \beta^\top x)\right) - \frac{\beta}{\sigma^2}\end{aligned}$$

We can see this expression has an exact closed form solution, as it does not depend on the intractable normalization constant.

(e) (2 points) Now let us use this score function $\nabla_\beta \log p(\beta \mid x, y)$ for sampling in Langevin MCMC.

i. Assume $x = 0$ and $y = 1$. Once the Langevin MCMC chain reaches its stationary distribution, what are the mean and variance of the samples?

ii. Assume $x = 1$ and $y = 1$. How does each term in the score function affect β?

Solution:

(i) If $x = 0$, the first term drops away and $\nabla_\beta \log p(\beta \mid x, y) = \nabla_\beta \log p(\beta)$. The stationary distribution of the Langevin MCMC chain is then the Gaussian prior $\beta \sim N(0, \sigma^2 I)$. Therefore, the mean of the samples is $\mathbb{E}[\beta] = 0$ and the covariance is $\mathrm{Var}[\beta] = \sigma^2 I$.

(ii) The first term, or the likelihood term, increases β but the gradient saturates to 0 as β goes to infinity. The second term, or the regularization term, penalizes large β.

initial here

7 (CS289A Only) Kernelized Nearest Neighbors (7 points)

Consider a dataset where each point X_i is a d-dimensional vector:

$$X_i = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}.$$

We would like to classify a query point X' using k-Nearest Neighbors (k-NN). However, instead of performing k-NN directly on the original features, we first transform the data using the function Φ that produces a quadratic embedding:

$$\Phi(X_i) = \begin{bmatrix} x_1^2 \\ \vdots \\ x_d^2 \\ x_1 x_2 \sqrt{2} \\ x_1 x_3 \sqrt{2} \\ \vdots \\ x_2 x_3 \sqrt{2} \\ \vdots \\ x_{d-1} x_d \sqrt{2} \\ x_1 \\ \vdots \\ x_d \end{bmatrix}.$$

(a) (2 points) In k-NN, we need to compute squared Euclidean distances between our query point X' and each point in our dataset. Fortunately, your friend Ruchir has found a kernel function k (not to be confused with the hyperparameter k in k-NN!) that works for the transformation Φ:

$$k(X_i, X_j) = (X_i^T X_j) + (X_i^T X_j)^2.$$

Write an expression for

$$||\Phi(X') - \Phi(X_i)||_2^2$$

only in terms of the kernel function k.

Solution:

$$\begin{aligned} ||\Phi(X') - \Phi(X_i)||_2^2 &= (\Phi(X') - \Phi(X_i))^\top(\Phi(X') - \Phi(X_i)) \\ &= \Phi(X')^\top\Phi(X') - 2\Phi(X')^\top\Phi(X_i) + \Phi(X_i)^\top\Phi(X_i) \\ &= k(X', X') - 2k(X', X_i) + k(X_i, X_i) \end{aligned}$$

(b) (2 points) Recall that X' and X_i are both d-dimensional. What is the big-O complexity of computing $\|\Phi(X') - \Phi(X_i)\|_2^2$ without kernelization? What is the complexity using the kernel Ruchir found? You do not have to show work.

Solution: No kernelization: $O(d^2)$. Kernelization: $O(d)$.

(c) (3 points) The RBF kernel is defined by

$$k(X_i, X_j) = \exp\left(-\frac{\|X_i - X_j\|_2^2}{2\sigma^2}\right).$$

For simplicity, we let $\sigma^2 = 1$ and $X_i \in \mathbb{R}$. Find a feature map Φ over an appropriate inner product space corresponding to this kernel. *Hint:* Recall the Taylor expansion:

$$e^X = \sum_{k=0}^{\infty} \frac{X^k}{k!}.$$

Solution: Consider the power series

$$P(x) = \exp\left(-\frac{x^2}{2}\right) \sum_{k=0}^{\infty} \frac{x^k}{\sqrt{k!}}$$

and the inner product $\langle P(x), P(y) \rangle$ defined by the sum of element-wise products of the power series (analogous to a dot product). Then we see that

$$\begin{aligned}\langle P(x), P(y) \rangle &= \exp\left(-\frac{x^2}{2}\right) \exp\left(-\frac{y^2}{2}\right) \sum_{k=0}^{\infty} \frac{x^k y^k}{k!} \\ &= \exp\left(-\frac{x^2}{2}\right) \exp\left(-\frac{y^2}{2}\right) \exp(xy) \\ &= \exp\left(-\frac{(x-y)^2}{2}\right) \\ &= k(x, y)\end{aligned}$$

Instead of writing the feature map as a power series, students can also choose to write it as an "infinitely long" bracket vector with the dot product as the inner product.

initial here

You may use this page to show extra work. Clearly mark your work with the problem number here, and also mention in the problem-specific box that your work is continued here.

initial here

You may use this page to show extra work. Clearly mark your work with the problem number here, and also mention in the problem-specific box that your work is continued here.

initial here

You may use this page to show extra work. Clearly mark your work with the problem number here, and also mention in the problem-specific box that your work is continued here.

initial here

You may use this page to show extra work. Clearly mark your work with the problem number here, and also mention in the problem-specific box that your work is continued here.