Solutions

10-315 Intro to ML
Spring 2023
Midterm Exam 2

Name: _____

Andrew ID: _____

Instructions:

- Please fill in your name and Andrew ID above.
- Be sure to write neatly and dark enough or you may not receive credit for your exam.
- This exam contains 22 pages (including this cover page and a blank page at the end). The total number of points is 100.
- Repeated from Piazza post: You use two two-sided, hand-written sheets of paper containing your notes to use during the exam. Note: this may NOT be created digitally and printed out. No calculators will be necessary or allowed.
- Note: All vectors on this exam are column vectors unless we state otherwise.
- Note: All logs are natural logs unless we state otherwise.

Below is a space for you to write any assumptions you make as you go through the exam that you would like us to consider. If you have nothing to state, you can leave this box blank.

Note: If at all possible, you should raise your hand to ask a clarifying question during the exam rather than relying on any assumptions.

1 Short Answer / Multiple Choice

1. (8 points) Which of the following machine learning algorithms are probabilistic generative models?

   Select all that apply.

   ☐ Logistic regression

   ☐ Logistic regression with ℓ_1 regularization

   ☐ Linear regression with a Gaussian prior on the parameters

   ☐ Convolutional neural network

   ☐ Naive Bayes with Categorical class distributions and Gaussian class-conditional distributions ← Answer

   ☐ Naive Bayes with Categorical class distributions and Bernoulli class-conditional distributions with add-one smoothing ← Answer

   ☐ Linear Discriminant Analysis ← Answer

   ☐ Quadratic Discriminant Analysis ← Answer

   ☐ None of the above

2. (2 points) True or False: Generative models allow us to formulate the joint distribution of the input features and the output, e.g. $p(y, x_1, x_2, x_3)$.

   ◯ True ◯ False True

3. (10 points) Probabilistic Models

   For each machine learning model, select the probabilistic model that it attempts to fit. Assume that $\mathbf{x}$ is the input, y is the output, and $\boldsymbol{\theta}$ is the vector of all parameters.

   (The set of options is the same for all of the following questions.)

   (a) Linear regression (without regularization)

   ◯ $p(y \mid \boldsymbol{\theta})$
   ◯ $p(\mathbf{x} \mid \boldsymbol{\theta})$
   ◯ $p(y \mid \mathbf{x}, \boldsymbol{\theta})$ ← Answer
   ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})$
   ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})$
   ◯ $p(y \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
   ◯ $p(\mathbf{x} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
   ◯ $p(y \mid \mathbf{x}, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
   ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
   ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$

   (b) Linear regression with L2 regularization

- ◯ $p(y \mid \boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid \boldsymbol{\theta})$
- ◯ $p(y \mid \mathbf{x}, \boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})$
- ◯ $p(y \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(y \mid \mathbf{x}, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$ ← Answer
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$

(c) Neural networks for classification (no regularization)

- ◯ $p(y \mid \boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid \boldsymbol{\theta})$
- ◯ $p(y \mid \mathbf{x}, \boldsymbol{\theta})$ ← Answer
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})$
- ◯ $p(y \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(y \mid \mathbf{x}, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$

(d) Classification using categorical class distributions and Gaussian class-conditional distributions. No additional assumptions including no naive Bayes assumption.

- ◯ $p(y \mid \boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid \boldsymbol{\theta})$
- ◯ $p(y \mid \mathbf{x}, \boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})$ ← Answer
- ◯ $p(y \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(y \mid \mathbf{x}, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$

(e) Classification using categorical class distributions and Gaussian class-conditional distributions with a naive Bayes assumption but no other assumptions.

- ◯ $p(y \mid \boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid \boldsymbol{\theta})$
- ◯ $p(y \mid \mathbf{x}, \boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})$ ← Answer
- ◯ $p(y \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(y \mid \mathbf{x}, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$
- ◯ $p(\mathbf{x} \mid y, \boldsymbol{\theta})\ p(y \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$

4. (4 points) Kernel Regression: Gammas and Lambdas

Consider kernel regression with an RBF filter, $k(\mathbf{x}, \mathbf{z}) = e^{-\gamma\|\mathbf{x}-\mathbf{z}\|_2^2}$, and the following process to train and predict:

1. **Step 1:** Compute $\boldsymbol{\alpha} = (K + \lambda I)^{-1}\mathbf{y}$ where $K_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ and $k(\mathbf{x}, \mathbf{z})$ is your kernel function.
2. **Step 2:** Given a new point $\mathbf{x}$, predict $\hat{y} = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}, \mathbf{x}^{(i)})$

Which of the following are likely to help reduce overfitting?

Select all that apply.

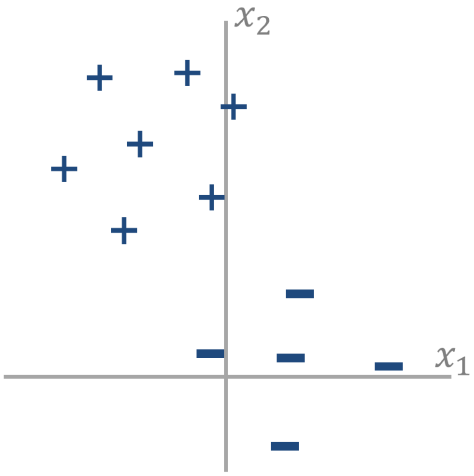
- ☐ Increase λ ← Answer
- ☐ Decrease λ
- ☐ Increase γ
- ☐ Decrease γ ← Answer
- ☐ None of the above

2 Regularization

1. We attempt to solve the binary classification task depicted in the following figure with the simple logistic regression model.

$$P(Y = 1 \mid \boldsymbol{x}, \boldsymbol{w}) = g(w_0 + w_1 x_1 + w_2 x_2) = \frac{1}{1 + \exp\left(-w_0 - w_1 x_1 - w_2 x_2\right)}$$

In the figure below, the + values represent class $Y = 1$, and the − values represent class $Y = 0$. Notice that the training data can be separated with zero training error with a linear separator.



(a) (6 points) Consider training a version of regularized logistic regression models where we try to maximize:

$$\sum_{i=1}^{N} \log P(y^{(i)} \mid \boldsymbol{x}^{(i)}, w_0, w_1, w_2) - C(w_j)^2$$

where hyperparameter $C >> 0$, and the penalty is on only one fixed j from $\{0, 1, 2\}$. In other words, only one of the parameters is penalized.

For the regularization of each w_j and $C >> 0$, state whether the training error on the above dataset increases, decreases, or stays the same when compared to the unpenalized logistic regression model. Provide a brief justification for each of your answers.

By penalizing w_0, the training error:

◯ Increases ◯ Remains the Same ◯ Decreases

Justification

Increases. When we penalize w_0, it will go to zero, The decision boundary will have to pass through the origin.

By regularizing w_1, the training error:

◯ Increases ◯ Remains the Same ◯ Decreases

Justification

Remains the same. When we regularize w_1, the resulting boundary can rely less and less on the values of x_1 and therefore becomes more horizontal and the training data can be separated with zero training error with a horizontal linear separator.

By regularizing w_2, the training error:

◯ Increases ◯ Remains the Same ◯ Decreases

Justification

Increases. When we regularize w_2, the resulting boundary can rely less and less on the value of x_2 and therefore becomes more vertical. For very large C, the training error increases as there is no good linear vertical separator of the training data.

(b) (3 points) If we change the form of regularization to L1-norm (absolute value) and regularize w_1 and w_2 only (but not w_0), we get the following penalized log-likelihood:

$\sum_{i=2}^{N} \log P(y^{(i)} \mid \boldsymbol{x}^{(i)}, w_0, w_1, w_2) - C(|w_1| + |w_2|)$

Consider again the problem in Figure 1 and the same logistic regression model $P(y = 1 \mid \boldsymbol{x}, \boldsymbol{w}) = g(w_0 + w_1 x_1 + w_2 x_2)$

As we increase the regularization parameter C, which of the following scenarios do you expect to observe? (Choose only one) Briefly explain your choice:

- ◯ First w_1 will become 0, then w_2
- ◯ First w_2 will become zero then w_1
- ◯ w_1 and w_2 will become zero simultaneously
- ◯ None of the weights will become exactly zero, only smaller as C increases.

Justification

The data can be classified with zero training error and therefore also with high log-probability by looking at the value of x_2 alone, i.e. making $w_1 = 0$. A non-zero value for w_1 is definitely better because it will allow for a better conditional likelihood, but w_1 will go to zero rather quickly as we increase regularization because we can still perfectly classify the data with $w_1 = 0$. Also, the absolute value regularization ensures that w_1 will indeed go to exactly zero. As C increases further, even w_2 will eventually become zero. We pay a higher and higher cost for setting w_2 to non-zero value. Eventually, this cost overwhelms the gain from the log-probability of labels that we can achieve with a non-zero w_2.

3 MLE and MAP

1. (6 points) Assume we have data $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$ and assume our $x^{(i)}$ are i.i.d from a distribution with the following density function: $f(x;\lambda) = \frac{\lambda^x}{x!}e^{-\lambda}$

   Write the expression for the likelihood of the data $\mathcal{D}$ in terms of $x^{(i)}$, λ and N?

$$\prod_{i=1}^{N} \frac{\lambda^{x^{(i)}}}{x^{(i)}!}e^{-\lambda} = \frac{\lambda^{\sum_i^N x^{(i)}}}{\prod_i^N x^{(i)}!}e^{-N\lambda}$$

   Write the expression for the log-likelihood of the data $\mathcal{D}$ in terms of $x^{(i)}$, λ and N?

$$\sum_i^N x^{(i)} \log(\lambda) - \log(\prod_i^N x^{(i)}!) - N\lambda$$

$$\sum_i^N x^{(i)} \log(\lambda) - (\sum_i^N \log x^{(i)}!) - N\lambda$$

   Derive the equation for the MLE of λ in terms of $x^{(i)}$ and N. You must show your work.

$$\frac{d\ell}{d\lambda} = \frac{\sum_i^N x^{(i)}}{\lambda} - N = 0$$

$$\implies \lambda = \frac{\sum_i^N x^{(i)}}{N}$$

2. Let binary Y be a random variable representing a coin flip. The probability of outcome heads, $Y = 1$ is modeled by a Bernoulli distribution with parameter $\phi \in [0, 1]$, i.e. $p(Y = 1 \mid \phi) = \phi$.

   Consider the observed outcomes of four coin flips being *tails*, *heads*, *tails*, *tails*, $\mathcal{D}_A = [y^{(1)}, y^{(2)}, y^{(3)}, y^{(4)}] = [0, 1, 0, 0]$.

   You may include basic arithmetic in your answers.

   (a) (4 points) MLE

   Write an expression for the likelihood for this dataset $p(\mathcal{D}_A \mid \phi)$ in terms of ϕ. Don't include any symbols for the data such as $\mathcal{D}_A$ or y.

   $p(\mathcal{D} \mid \phi)$:

   $\phi(1 - \phi)^3$

   For this data, $\mathcal{D}_A$, what is the maximum likelihood estimate of ϕ?

   $\hat{\phi}_{\text{MLE}}$:

   0.25

   (b) (6 points) MAP

   We now add a prior on the parameter ϕ, specifically $p(\phi) = 2\phi$.

   Write an expression for $p(\mathcal{D}_A, \phi)$ in terms of ϕ. Don't include any symbols for the data such as $\mathcal{D}_A$ or y.

   $p(\mathcal{D}, \phi)$:

   $\phi(1 - \phi)^3 2\phi = 2\phi^2(1 - \phi)^3$

   Write an expression for the objective function $J(\phi) = -\log p(\mathcal{D}_A, \phi)$ in terms of ϕ. Don't include any symbols for the data such as $\mathcal{D}_A$ or y.

   $J(\phi) = -\log p(\mathcal{D}, \phi)$:

   $-\log 2 - 2\log\phi - 3\log(1-\phi)$

Write an expression for the derivative of the objective function with respect to ϕ, $dJ/d\phi$.

$dJ/d\phi$:

$-2/\phi + 3/(1-\phi)$

For this data, $\mathcal{D}_A = [0, 1, 0, 0]$, what is the MAP estimate of ϕ?

$\hat{\phi}_{\text{MAP}}$:

0.4

(c) (4 points) MAP with more points

Now, we flip the coin a total of 99 times and end up with dataset $\mathcal{D}_B$ containing 11 heads and 88 tails.

Using the same prior as before, $p(\phi) = 2\phi$, compute the MAP estimate for this new dataset, $\mathcal{D}_B$.

$\hat{\phi}_{\text{MAP}}$:

0.12

Finally, we flip the coin a total of 999 times and end up with dataset $\mathcal{D}_C$ containing 111 heads and 888 tails.

Using the same prior as before, $p(\phi) = 2\phi$, compute the MAP estimate for this new dataset, $\mathcal{D}_C$.

$\hat{\phi}_{\text{MAP}}$:

0.112

4 Naive Bayes

1. (8 points) We are given a database of vehicles and their features with a theft record of whether they were stolen or not.

Example	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Use Bernoulli naive Bayes without smoothing to compute the following probabilities given this dataset.

Note: Leave your answers as unsimplified arithmetic expressions, i.e. you don't have to simplify your arithmetic down to numerical value. Please show any work.

Given a new car $\mathbf{x} = [Red, SUV, Domestic]$, what is $P(Stolen = Yes, \mathbf{x})$?

$$P(Stolen = Yes, \mathbf{x}) = \frac{5}{10}\frac{3}{5}\frac{1}{5}\frac{2}{5} = \frac{3}{125} = 0.024$$

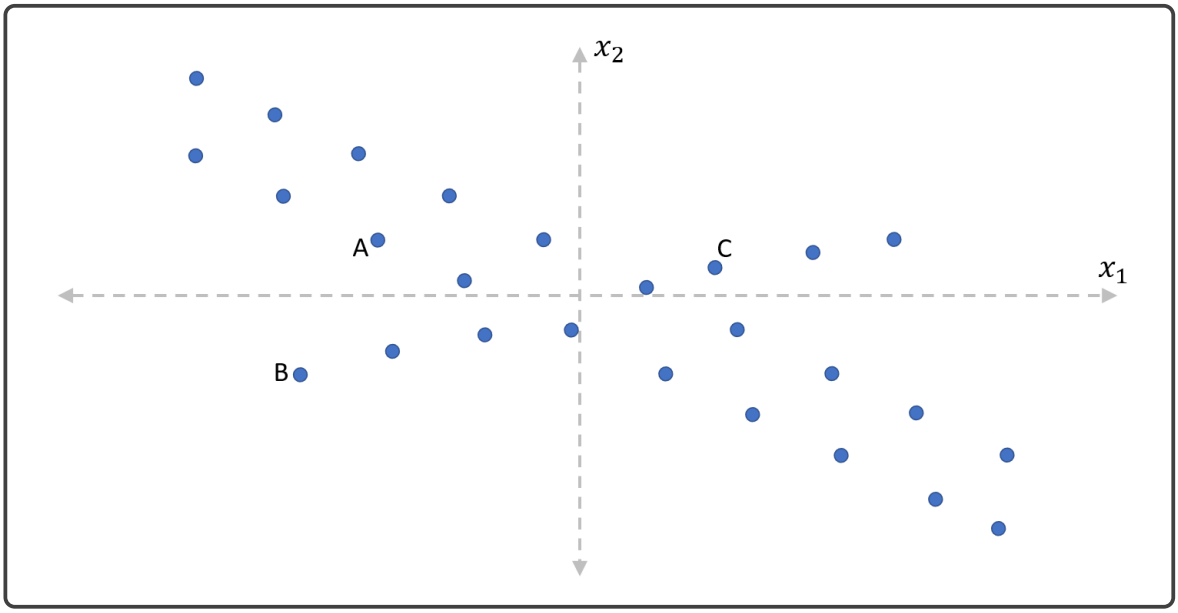
Given the same new car $\mathbf{x} = [Red, SUV, Domestic]$, what is $P(Stolen = Yes \mid \mathbf{x})$?

$$P(Stolen = No, \mathbf{x}) = \frac{5}{10}\frac{2}{5}\frac{3}{5}\frac{3}{5} = \frac{9}{125}$$

$$P(Stolen = Yes \mid \mathbf{x}) = \frac{\frac{3}{125}}{\frac{3}{125} + \frac{9}{125}} = \frac{1}{4}$$

5 PCA

1. (6 points) On the figure below, draw the following:
   - The first principal component: as an arrow labeled 1
   - The second principal component: as an arrow labeled 2
   - The reconstructed points for the 3 data points labeled A, B, C after projecting onto the first principal component
   - The reconstruction error lines connecting points A, B, C and their reconstructed points after projecting onto the first principal component



- First principal component at roughly 30 degree angle below positive x_1 axis (or above negative x_1 axis
- Second principal component must be perpendicular to first
- Reconstructed points should be on first principal component and error lines should be perpendicular to the first principal component and connected to the original points.

6 Recommender Systems

1. (3 points) Given a sparse set of training recommendations, $r_{i,j}$, for N users and M items, assume that you have already solved for the matrix factorization of $R \in R^{N \times M}$ and now have matrices $U \in R^{N \times K}$ and $V \in R^{M \times K}$.

   Now, you are given a new user, user $N+1$, that was not considered at all during your training. You collect reviews from that user on the first 5 items. Describe precisely how you would go about recommending the single item that the user is most likely to rate the highest (other than the first 5 items of course).

   You'd have to:
   1) add your ratings to R and add a row to U, 2) retrain, and 3) take the $\arg\max_j u_{N+1}^T v_j$.
   It would probably be beneficial to initialize the new system with the previous U and V.

2. (6 points) Consider the following ratings matrix, where '?' indicates that no rating has been given:

$$R = \begin{bmatrix} 4 & ? & ? \\ ? & 2 & ? \\ ? & 5 & ? \\ ? & ? & 3 \end{bmatrix}$$

We would like to use matrix factorization to embed our users and items in a $K = 2$ dimensional space. Write specific U and V matrices that optimize the matrix factorization objective function $J(U, V)$ for $K = 2$:

$$J(U, V) = \frac{1}{2} \sum_{i,j \in \mathcal{S}} (r_{i,j} - \mathbf{u}_i^T \mathbf{v}_j)^2$$

where, as usual, $\mathbf{u}_i^T$ is the i-th row of U, $\mathbf{v}_j^T$ is the j-th row of V, and $\mathcal{S}$ is the set of indices, i, j, for ratings that are not '?'.

Note: There is more than one correct answer.

Write your answers as matrices filled with numerical values.

U:

V:

The dot product of the first row of U and the first row of V must equal 4.

The dot product of the second row of U and the second row of V must equal 2.

The dot product of the third row of U and the second row of V must equal 5.

The dot product of the fourth row of U and the third row of V must equal 3. For example, $U = \begin{bmatrix} 4 & 0 \\ 2 & 0 \\ 5 & 0 \\ 3 & 0 \end{bmatrix}$ and $V = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$

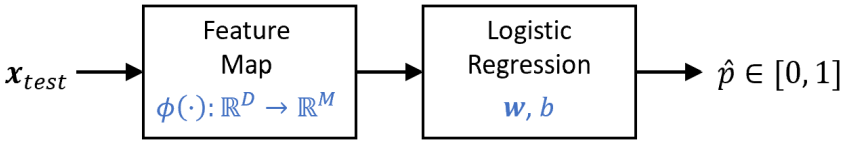
7 Applying ML

Once a machine learning system has been trained, it is important to understand how to use it. For the following trained systems, give a precise mathematical equation to compute the output for a given input.

If you would like to use any temporary variables, be sure to give the equation for that variable AND define the size of the variable, e.g. scalar or $\in R^N$, etc.

We'll start with an example, so you know what to expect.

Logistic Regression (example)



System: We applied a feature map to our input data, $\phi(\cdot)$, and then trained a binary logistic regression model on the mapped features. The logistic regression model included a bias term.

Test: For a new input, $\mathbf{x}_{test}$, give an equation for $\hat{p}$, the predicted probability of $\mathbf{x}_{test}$ belonging to class $Y = 1$.

Example acceptable answer:

$\hat{p} = \frac{1}{1+e^{-(w^T \phi(x_{test})+b)}}$

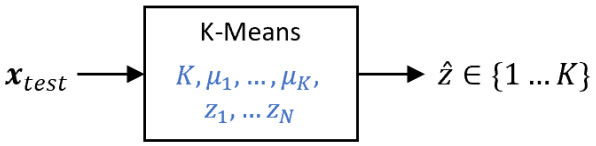
Example insufficient answers:

$\hat{p} = g(w^T \phi(x_{test}) + b)$ // Issue: Using undefined function g

$\hat{p} = \frac{1}{1+e^{-z}}$ // Issue: Using undefined variable z

$\hat{p}$ = the logistic function of the weight vector dotted with the feature map vector plus the bias term // Issue: Using words, not precise math

1. (3 points) **K-means**

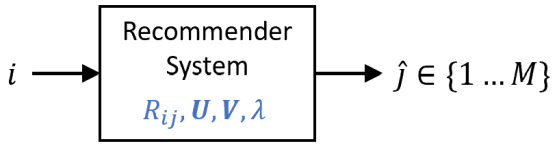


System: We clustered unlabelled input data $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ into K clusters using K-means. K-means produced cluster centers, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ and cluster assignments, $z_1, \ldots, z_N$.

Test: For a new input, $\mathbf{x}_{test}$, give an equation for $\hat{z}$, the predicted cluster assignment for $\mathbf{x}_{test}$.

$\hat{z} = \text{argmin}_j \|\mathbf{x}_{test} - \boldsymbol{\mu}_j\|_2$

2. (3 points) **Recommender Systems**



System: We solved the following regularized matrix factorization optimization for a given sparse set of ratings $\{R_{ij}\}_{i,j\in\mathcal{S}}$, where $\mathcal{S}$ is the set of all pairs (i, j) where there exists a rating from the i-th user on the j-th item.

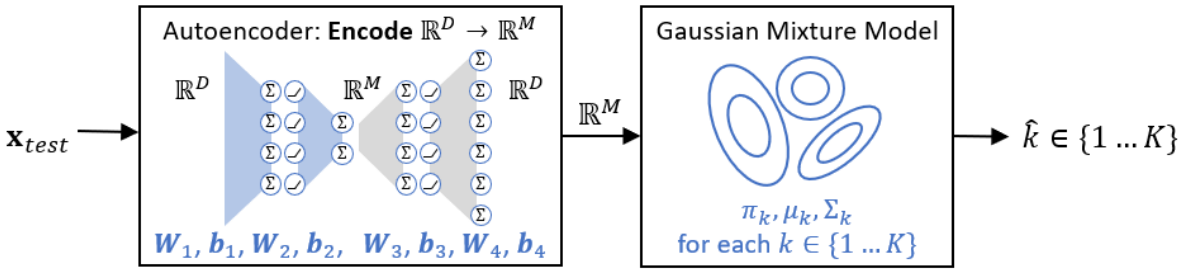
$\text{argmin}_{\mathbf{U},\mathbf{V}} \sum_{i,j\in\mathcal{S}}(R_{ij} - \mathbf{u}_i^T\mathbf{v}_j)^2 + \lambda(\sum_i ||\mathbf{u}_i||_2^2 + \sum_j ||\mathbf{v}_j||_2^2)$

For hyperparameter λ, the optimization returned an $N \times K$ user matrix, $\mathbf{U}$, and an $M \times K$ item matrix, $\mathbf{V}$, where $\mathbf{u}_i^T$ is the i-th row of $\mathbf{U}$ and $\mathbf{v}_j^T$ is the j-th row of $\mathbf{V}$.

Test: For user index i, give an equation for $\hat{j}$, the index with the highest predicted rating for that user. To simplify things, it is ok if this predicted index is of an item that user i already rated.

$\hat{j} = \arg\max_j \mathbf{u}_i^T\mathbf{v}_j$

3. (6 points) **Grouping Photos**



System: Identify which cluster an input image is likely to belong to.

To do this, we first trained an autoencoder network to map the D number of (vectorized) input pixels into an M-dimensional space. Second, we trained a Gaussian mixture model with K classes on the encoded M-dimensional points (not the D-dimensional recovered points). Finally, we use the GMM to determine the index, $\hat{k}$, of the cluster that has the highest probability of the encoded test point belonging to that cluster.

The neural network (encoder then decoder) has a total of four fully-connected layers. The first and third layers use a ReLU activation function; the second and output layers have no activation functions. The trained autoencoder network contains the following parameters: $W_1 \in \mathbb{R}^{L \times D}$, $\mathbf{b}_1 \in \mathbb{R}^L$, $W_2 \in \mathbb{R}^{M \times L}$, $\mathbf{b}_2 \in \mathbb{R}^M$, $W_3 \in \mathbb{R}^{L \times M}$, $\mathbf{b}_3 \in \mathbb{R}^L$, $W_4 \in \mathbb{R}^{D \times L}$, $\mathbf{b}_4 \in \mathbb{R}^D$.

The Gaussian mixture model has a set of parameters for each class: mixture probabilities, means, and covariance matrices, $\pi_k \in [0, 1]$, $\boldsymbol{\mu}_k \in \mathbb{R}^M$, and $\Sigma_k \in \mathbb{R}^{M \times M}$, respectively.

Test: For an input image in vector form, $\mathbf{x}_{test}$, give an equation for $\hat{i}$, the index of the training image that matches bestswaqq.

You may use the following functions in your answer:

- $ReLU(\mathbf{z})$, which applies a ReLU to each entry in the input vector, $\mathbf{z}$, and returns the resulting output vector
- $f(\mathbf{z}, \boldsymbol{\mu}, \Sigma)$, which returns the density from the multidimensional Gaussian pdf at $\mathbf{z}$

Note: Given the length of the answer, you should temporary variables for this question. Be sure to give the equation for that variable AND define the size of the variable, e.g. scalar or $\in R^N$, etc.

Define $\mathbf{v}_{test} \in \mathbb{R}^M$
$\mathbf{v}_{test} = W_2 ReLU(W_1 \mathbf{x}_{test} + \mathbf{b}_1) + \mathbf{b}_2$
$\hat{k} = \text{argmax}_k \ \ \pi_k f(\mathbf{v}_{test}, \mu_k, \Sigma_k)$

8 Word Embeddings

1. Consider the following word2vec SGD objective function for a single pair of tokens in a training corpus with N tokens, where (i, j) is the corresponding pair of indices into a vocabulary of M tokens:

$$J(U, V)^{(i,j)} = -\log g_{softmax}\left(V\mathbf{u}_i\right)$$

where $g_{softmax}$ is the softmax function, $\mathbf{u}_i$ is the i-th row of U as a column vector, and $\mathbf{v}_j$ is the j-th row of V as a column vector.

Let's define the following:

- $\mathbf{s} = V\mathbf{u}_i$, the score vector for each vocab token
- $\hat{\mathbf{y}} = g_{softmax}\left(\mathbf{s}\right)$, the predicted probability for each vocab token
- $\mathbf{y}$ is the one-hot vector with a one at index j.

The following are the corresponding SGD gradient update expressions, derived using the partial derivative of the cross-entropy and softmax functions, $\partial J/\partial \mathbf{s} = \hat{\mathbf{y}} - \mathbf{y}$:

$$\mathbf{u}_i \leftarrow \mathbf{u}_i - \alpha V^\top \left(\hat{\mathbf{y}} - \mathbf{y}\right)$$
$$V \leftarrow V - \alpha \left(\hat{\mathbf{y}} - \mathbf{y}\right) \mathbf{u}_i^\top$$

(a) (3 points) Give your answers to the following in terms of N, M, K, where K is the number of dimensions of embedded space.

What is the size of U:

$M \times K$

What is the size of V:

$M \times K$

How many iterations are in one epoch of SGD (batch size one)?

$N - 1$

2. Now, we would like to train a word2vec model to predict the next word from a context of exactly $C = 3$ previous words rather than just one, as in the model above.

   (a) (2 points) How many parameters are in this model in terms of N, M, K?

   $2MK$

   How many iterations are in one epoch of SGD (batch size one)in terms of N, M, K?

   $N - 3$

   (b) (2 points) Write the equation for the vector $\hat{\mathbf{y}}$ containing predicted next token probabilities for each token in the vocabulary, given input context indices, i_1, i_2, i_3. You may use $g_{softmax}$ in your answer.

   $$\hat{\mathbf{y}} = g_{softmax}\left(V\left(\frac{1}{3}\sum_{c=1}^{3}\mathbf{u}_{i_c}\right)\right)$$

   (c) (2 points) Write the SGD objective function for four consecutive tokens in the training corpus where i_1, i_2, i_3, i_4 are the corresponding vocabulary indices, where i_4 is the index of the fourth word. You may write your answer in terms of your definition for $\hat{\mathbf{y}}$ above.

   $$J(U, V)^{(i,j)} = -\log \hat{\mathbf{y}}_{i_4}$$

(d) (3 points) Write the SGD gradient update expressions for one tuple of vocabulary indices, i_1, i_2, i_3, i_4, and learning rate α. You may write your answer in terms of your definition for $\hat{\mathbf{y}}$ above as well as the one-hot vector $\mathbf{y}$ that has a one at index i_4.

$$\mathbf{u}_{i_1} \leftarrow \mathbf{u}_{i_1} - \alpha \frac{1}{3} V^\top (\hat{\mathbf{y}} - \mathbf{y})$$

$$\mathbf{u}_{i_2} \leftarrow \mathbf{u}_{i_2} - \alpha \frac{1}{3} V^\top (\hat{\mathbf{y}} - \mathbf{y})$$

$$\mathbf{u}_{i_3} \leftarrow \mathbf{u}_{i_3} - \alpha \frac{1}{3} V^\top (\hat{\mathbf{y}} - \mathbf{y})$$

$$V \leftarrow V - \alpha (\hat{\mathbf{y}} - \mathbf{y}) \left(\frac{1}{3} \sum_{c=1}^{3} \mathbf{u}_{i_c}^\top \right)$$

This page intentionally left blank.