CPSC 340 2021S Midterm Exam - Written Portion (Real Thing)

IMPORTANT: Download the .tex source file here!!!!

https://www.students.cs.ubc.ca/~cs-340/namhee/midterm_exam.tex

Instructions

As with the assignments, please indicate your name and 8-digit UBC student ID number in your solution.

- You have 55 minutes to complete this portion of the exam. **Recommended: finish in 45 minutes and use the last 10 minutes to submit on Gradescope.**
- You may submit a handwritten solution instead of a typeset document. If you do so, please clearly mark your answers with corresponding question numbers.
- You can do this exam anytime during the 24-hour midterm exam window.
- The exam is open book, meaning you are allowed to consult course materials, the internet, etc.
- You may NOT communicate with anyone else (other than the instructor) in any way during the exam. This includes posting anything on the internet (e.g. **Discord**, a message board, chat, tutoring service, etc.) during the exam. UBC has a strict policy on academic misconduct, along with disciplinary measures, which I can attest are not fun for anyone involved.
- You may NOT copy/paste code from anywhere. All code submitted should be code that you wrote during the exam. If you consulted any online resources and wish to cite this, feel free to drop a link in your exam submission to be safe.
- Announcements or clarifications will be made on this Piazza thread:
  https://piazza.com/class/koaczotaebx55t?cid=194.
  Please check it before the exam.
- If you have a question, make a private post on Piazza.
- Do not post or communicate about the exam until at least 1 hour after the scheduled end time.
- This is NOT the entire midterm exam - there is also a Canvas portion!
- The written portion and Canvas portion are **equally weighted**.

The submission format for this portion of the midterm exam is identical to the submission format for the homework assignments, except two things: (1) you cannot work with a partner, and (2) you can submit handwritten solutions.

Rubric: {points:5}

The above points are allocated for following the official submission instructions. In short, submit to Gradescope and match the questions to pages when prompted by Gradescope.

1 Very-Short Answer Questions

Rubric: {points:20}

1. Recall summary statistics. Describe one scenario where mean is a poor measure of average.

   Answer: When there are outliers in the data, or the data is not unimodal.

2. Recall using naïve Bayes for spam filtering. Describe a disadvantage of having a non-binary bag-of-words for representing emails when using a probabilistic classifier (i.e. not necessarily naïve Bayes).

   Answer: We need more examples to observe at least one example per possible feature combination.

3. Recall k-nearest neighbours. Suppose I have $d = 100,000$ features and $n = 100$ training examples. Describe one reason why my test error may be high regardless of the value of k.

   Answer: The curse of dimensionality. The test example's nearest neighbours in the training set might not have any meaningful similarity.

4. Recall vector quantization based on k-means clustering. Assuming every example is unique, what is the value of k that would make vector quantization completely lossless (i.e. we retain 100% of the information in the feature matrix X)?

   Answer: When $k = n$, then every example has its own cluster. So vector quantization will be 100% lossless but will not compress anything.

5. Recall hierarchical clustering, constructed with the agglomerative clustering algorithm. Is hierarchical clustering a parametric or non-parametric method? Explain why.

   Answer: Non-parametric. The depth of the tree will tend to grow with more examples. The memory complexity will grow because there are more leaf nodes to store with more examples.

6. Recall the normal equations for solving the ordinary least squares problem for linear regression. Describe a scenario where we cannot express the solution w as a product of matrices and vectors.

   Answer: When X^TX is not invertible, i.e. colinearity in features of X.

7. Recall gradient descent. Describe why gradient descent may be necessary for minimizing an objective function, even when the number of features d is relatively small.

   Answer: The minimizers may not have closed-form solutions.

8. Recall Huber loss for robust regression. Describe the effect of increasing the variable ϵ on the model's robustness to outliers.

   Answer: When ϵ increases, the model will behave more like ordinary least squares, hence robustness decreases.

9. Suppose a new classification model called Gelbus-2021 (G2021) was invented. G2021 randomly splits the data into m non-overlapping subsets. To classify a new test example $\tilde{x}_i$, G2021 looks at $\tilde{x}_i$'s nearest neighbour in each subset to get m labels, and returns the most common value among the m labels. Describe the effect of the hyper-parameter m on the two parts of the fundamental trade-off.

   Answer: When $m = 1$, then this is same as k-nearest neighbours with $k = 1$. When $m = n$, then this is the same as return-the-mode. Therefore, as m increases, training error will increase but approximation error will decrease.

10. Is G2021 a parametric or non-parametric model? Explain why.

    Answer: Non-parametric. We need to store the entire dataset, just as with k-nearest neighbours.

2 k-Nearest Neighbours

Rubric: {points:10}

Suppose I have the following training data with continuous features and categorical labels:

$$X = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 0 \\ 3 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad y = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

1. Suppose I have a test example $\tilde{x}_i = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$. According to the k-nearest neighbours classifier with $k = 5$ using L_1-norm of difference as the distance function, what is the predicted label $\hat{y}_i$?

   Answer: $\hat{y}_i = 1$. We don't even need to look at the features because $k = n$.

2. Suppose I have a test example $\tilde{x}_i = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$. According to the k-nearest neighbours classifier with $k = 1$ using L_∞-norm of difference as the distance function, what is the predicted label $\hat{y}_i$?

   Answer: $\hat{y}_i = 4$. $\max_j |\tilde{x}_{ij} - x_{ij}|$ for each row is: 2, 2, 2, 3, 1. So the last row is the nearest neighbour.

3 Memory Complexity Analysis

Rubric: {points:10}

Answer the following questions using the big-O notation.

1. What is the amount of memory needed for an ensemble classifier of m submodels, where the submodels are k-nearest neighbours classifiers with $k = 1, 2, \cdots, m$? Your answer may or may not depend on n, d, m, and k.

   Answer: $O(nd)$. Even though we have an ensemble of KNNs, we only need to ship the same training data.

2. Recall that the parameters of the naïve Bayes classifier are the frequencies corresponding to $p(y_i)$ and $p(x_{ij} \mid y_i)$. What is the memory required for training the naïve Bayes classifier on a dataset where each feature x_{ij} is a categorical feature that can have m values and the label y_i can have k classes? You answer may or may not depend on n, d, m, and k.

   Answer: $O(dmk)$. We need $O(k)$ for storing k counts for $p(y_i)$. For each feature j, we need to store $p(x_{ij} = 0 \mid y_i = c), p(x_{ij} = 1 \mid y_i = c), \cdots, p(x_{ij} = m \mid y_i = c)$, across all values of c. Then for each feature j, we need $O(mk)$ memory, so we need $O(dmk)$ memory total.

4 Objective Function

Rubric: {points:15}

See the "non-negative least squares" objective function below (assume $\lambda > 0$):

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w^T x_i - y_i)^2 + \lambda \sum_{j=1}^{d} \max\{0, -w_j\}$$

1. Look at the below statement. Remove the incorrect option in each parenthesis so that the resulting sentence is true:

   Answer: This f is (**convex** / non-convex), (**continuous** / non-continuous), and (differentiable / **non-differentiable**).

2. Your answer to the previous question should make you realize that a certain kind of approximation is necessary for gradient-based methods. Re-write f such that it incorporates this approximation.

   Answer:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w^T x_i - y_i)^2 + \lambda \sum_{j=1}^{d} \log(1 + \exp(-w_j))$$

3. Write down the gradient $\nabla f(w)$. You do **not** need to express this in matrix and vector forms, although it's possible.

   Answer: For simplicity, let me call the first term "left" and the second term "right".

$$f(w) = \text{left} + \text{right}$$

"left" is exactly the same as ordinary least squares error. Gradient of "left" is $X^T Xw - X^T y$.
As with the practice midterm, "right" is a little tricky. The easiest way is to take the partial derivative with respect to a particular w_j.

$$\begin{aligned}
\frac{\partial}{\partial w_1}\text{right} &= \lambda \frac{\partial}{\partial w_1}\sum_{j=1}^{d}\log(1+\exp(-w_j)) && (1)\\
&= \lambda \frac{\partial}{\partial w_1}\log(1+\exp(-w_1)) && (2)\\
&= \lambda \frac{1}{1+\exp(-w_1)} \cdot \exp(-w_1) \cdot -1 && (3)\\
&= -\lambda \frac{\exp(-w_1)}{1+\exp(-w_1)} && (4)
\end{aligned}$$

It's completely fine to leave the answer as partial derivative with respect to some j, i.e.

$$\frac{\partial}{\partial w_j} f(w) = \sum_{i=1}^{n}(w^T x_i - y_i)x_{ij} - \lambda \frac{\exp(-w_j)}{1+\exp(-w_j)} \quad (5)$$

With that said, we can create a $d \times 1$ vector r, such that

$$r = \begin{bmatrix} \frac{\exp(-w_1)}{1+\exp(-w_1)} \\ \frac{\exp(-w_2)}{1+\exp(-w_2)} \\ \vdots \\ \frac{\exp(-w_d)}{1+\exp(-w_d)} \end{bmatrix} \quad (6)$$

Then the final gradient is

$$\begin{aligned}
\nabla f &= \nabla\text{left} + \nabla\text{right} && (7)\\
&= X^T Xw - X^T y - \lambda r. && (8)
\end{aligned}$$

One could also try computing the non-smooth gradient in the original f, so the "right" term in the gradient look like

$$\frac{\partial}{\partial w_j}\text{right} = \begin{cases} -1 & \text{if } w_j \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$