CPSC 340 2021S Midterm Exam - Written Portion (Practice)
45 Minutes

IMPORTANT: Download the source file here!!!!
`asdf`

Instructions

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

- 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. 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. Submitting copied code **is considered academic misconduct.**
- Announcements or clarifications will be made on this Piazza thread:
`https://piazza.com/class/koaczotaebx55t?cid=188`.
Please check it occasionally during 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 coding 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 include handwritten solutions as figures.

The above points are allocated for following the general homework instructions on the course homepage. In short, submit to Gradescope and match the questions to pages when prompted by Gradescope.

1 Very-Short Answers

Rubric: {points:26}

1. Recall feature transformations. Describe one way to convert categorical features (e.g. Vancouver, Surrey, Richmond) into numerical features.

   Answer: One-hot encoding.

2. Recall the validation set approach. Describe a disadvantage of splitting your training data into 99% training examples and 1% validation examples. Describe a disadvantage of splitting your training data into 1% training examples and 99% validation examples.

   Answer: Small number of validation examples makes optimization bias easier to occur. Small number of training examples makes it hard to learn.

3. Recall decision theory. Describe an example where increasing false negatives is an acceptable risk.

   Answer: Anything where we want to minimize false positives, e.g. spam filtering.

4. Recall k-means clustering. Describe a reason why using an ensemble of k-means clustering may not be a good idea.

   Answer: Label switching across different k-means models will make the $\hat{y}_i$ values meaningless.

5. Recall density-based clustering and k-means clustering. In terms of outlier detection, describe a reason why density-based clustering might be better than k-means clustering.

   Answer: Density-based clustering will detect outliers with respect to non-convex shapes. Density-based clustering will also explicitly leave unlabeled examples, which can be considered outliers.

6. Recall using the Z-score for outlier detection. Describe one scenario where Z-score may be a good choice for outlier detection.

   Answer: When we have a uni-modal data, i.e. we have one clear "center" for the examples, and when outliers don't shift the mean of the normal distribution.

7. Recall classification accuracy $\sum_{i=1}^{n}(y_i = \hat{y}_i)$. Describe why classification accuracy is not a good measure of error for regression problems.

   Answer: For regression, y_i values are continuous and thus $\hat{y}_i = y_i$ will never occur in practice.

8. Recall change of basis and nonlinear regression. Suppose we are performing a polynomial basis feature transform. Describe one reason why a large degree of polynomial may be undesirable.

   Answer: A large degree of polynomial can lead to overfitting.

9. Recall gradient descent. Suppose two different runs of gradient descent return two different minimizers. Describe one reason why this might happen.

   Answer: The function can be non-convex, or convex with a flat region of local minima.

10. Recall robust and brittle regressions. Describe a scenario where brittle regression is more appropriate than robust regression.

    Answer: When we want to get accurate predictions for outliers while sacrificing accuracy on non-outliers.

11. Suppose a new supervised learning algorithm called Markus-2021 (M2021) was invented. M2021 lets the user specify P, the number of parameters in the model. Given the same number of training examples, we observe that a larger number of parameters makes M2021 more complex. What is the effect of P on the two parts of the fundamental trade-off?

    Answer: Increasing P means increased model complexity. Therefore, as P increases, training error goes down and approximation error goes up.

12. Suppose I specified $P = 10^9$ (1 billion) for M2021 to fit X and y, where $n = 10^3$ (1 thousand). I get a training error of 0, but I suspect that this model is severely overfitting. How can I check whether my model is overfitting?

    Answer: Use the validation set approach.

13. I want to choose a good value of P for M2021 that minimizes the test error. So I do a k-fold cross validation with $k = n$ and search over possible values of P, which is all integers ranging from 1 to 10^9. What is wrong with this approach?

    Answer: It will take forever.

2 Naïve Bayes

Rubric: {points:15}

Consider the dataset below, which has 5 training examples, 2 features, and 3 classes.

$$X = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \quad y = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \\ 2 \end{bmatrix} \tag{1}$$

1. According to Naïve Bayes, what is the probability $P(x_i = [1 \quad 1]^T \mid y_i = 0)$?

   Answer: Using conditional independence,

$$\begin{aligned} P(x_i = [1 \quad 1]^T \mid y_i = 0) &= P(x_{i1} = 1 \mid y_i = 0)P(x_{i2} = 1 \mid y_i = 0) && (2)\\ &= 1/2 \cdot 1/2 && (3)\\ &= 1/4 && (4) \end{aligned}$$

2. According to Naïve Bayes, what is the label $\hat{y}_i$ for $\hat{x}_i = [1 \quad 1]^T$?

   Answer: We should calculate the numerators of the three conditional probabilities:

$$\begin{aligned} P(y_i = 0 \mid x_i = [1 \quad 1]^T) &\propto P(x_i = [1 \quad 1]^T \mid y_i = 0)P(y_i = 0) && (5)\\ &= P(x_{i1} = 1 \mid y_i = 0)P(x_{i2} = 1 \mid y_i = 0)P(y_i = 0) && (6)\\ &= 1/2 \cdot 1/2 \cdot 2/5 = 1/10 && (7)\\ P(y_i = 1 \mid x_i = [1 \quad 1]^T) &\propto P(x_i = [1 \quad 1]^T \mid y_i = 1)P(y_i = 1) && (8)\\ &= P(x_{i1} = 1 \mid y_i = 1)P(x_{i2} = 1 \mid y_i = 1)P(y_i = 1) && (9)\\ &= 1 \cdot 1 \cdot 1/5 = \mathbf{1/5} && (10)\\ P(y_i = 2 \mid x_i = [1 \quad 1]^T) &\propto P(x_i = [1 \quad 1]^T \mid y_i = 2)P(y_i = 2) && (11)\\ &= P(x_{i1} = 1 \mid y_i = 2)P(x_{i2} = 1 \mid y_i = 2)P(y_i = 2) && (12)\\ &= 1/2 \cdot 1/2 \cdot 2/5 = 1/10 && (13) \end{aligned}$$

   According to these probabilities, we should choose $\hat{y}_i = 1$.

3. According to Naïve Bayes **with Laplace smoothing where** $\beta = 1$, what is the probability $P(x_i = [1 \quad 1]^T \mid y_i = 0)$?

   Answer:

$$\begin{aligned} P(x_i = [1 \quad 1]^T \mid y_i = 0) & && (14)\\ &= P(x_{i1} = 1 \mid y_i = 0)P(x_{i2} = 1 \mid y_i = 0) && (15)\\ &\approx (1+1)/(2+3) \cdot (1+1)/(2+3) = 2/5 \cdot 2/5 && (16)\\ &= 4/25 && (17) \end{aligned}$$

3 Time Complexity Analysis

Rubric: {points:10}

1. What is the time complexity of predicting the label of t test examples using the k-nearest neighbors algorithm when $k = n$? Your answer may (or may not) depend on t, n, d, k.

   Answer: $O(n + t)$. If $k = n$, this is exactly the same thing as return-the-mode, so we do not need to look at the features at all. We compute the mode in $O(n)$ and return this for every test example in $O(1)$ time. Partial credit for $O(nt)$.

2. Suppose I have a model, whose training time complexity is $O(nd)$ and testing time complexity is $O(td)$. Suppose I have h different hyper-parameters, whose values are either 0 or 1. What is the time complexity of finding the optimal values of the hyper-parameters using the validation set approach, where we use 50% of training data as validation data? Your answer may (or may not) depend on 2, n, d, t, h.

   Answer: $O(2^h nd)$. For each hyperparameter vaule, we train on $\frac{n}{2}$ examples and and evaluate on $\frac{n}{2}$ examples, so we need $O(nd)$ time to get a validation error. There are $O(2^h)$ different hyperparameter values to try.

4 Objective Function

Rubric: {points:20}

See the objective function below (assume $\lambda > 0$):

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

1. Write down the gradient $\nabla f(w)$ using matrix and vector forms. You may use the element-wise sign function `sgn()`.

   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 $X^T Xw - X^T y$.
   "right" is a little tricky, but thinking in terms of partial derivatives, we can tackle it.

$$\frac{\partial}{\partial w_1}\text{right} = \lambda \frac{\partial}{\partial w_1}\sum_{j=1}^{d}|w_j| \tag{18}$$

$$= \lambda \frac{\partial}{\partial w_1}|w_1| \tag{19}$$

$$= \lambda \begin{cases} 1 & \text{if} \quad w_1 \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

Equation (20) is easier to see if you plot the function $y = |x|$. When we stack these partial derivatives together, we see that $\nabla \text{right} = \texttt{sgn}(w)$.
Therefore,

$$\nabla f = \nabla \text{left} + \nabla \text{right} \tag{21}$$
$$= X^T X w - X^T y + \texttt{sgn}(w). \tag{22}$$

2. 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**). Therefore, we (can / **can't**) use gradient descent for this objective function.