DS-GA-1003: Machine Learning (Spring 2022)

Final Exam (6:00pm–8:00pm, May 12)

- You should finish the exam within **1 hours and 45 minutes** and submit through Gradescope by **8pm**.
- You can refer to textbooks, lecture slides, and notes. However, searching answers online and collaborating with others are not allowed.
- Please write your solution on a separate sheet or the released exam sheet clearly, then upload the photo of your handwriting or the annotated pdf to Gradescope.
- **Make sure you leave enough time to upload the exam. Gradescope will terminate the submission window at 8:00pm promptly.** There is no grace period.

Question	Points	Score
Probabilistic models	14	
Bayesian methods	14	
Multiclass classification	15	
Decision trees and ensemble methods	12	
Boosting	11	
Neural networks	19	
Clustering	15	
Total:	100	

1. **Help the TAs!** Vishakh and Colin are trying to model the time it takes to answer a student's question during office hour so that they can better plan ahead. You will help them build a model to do this. One common distribution to model the duration of independent events is the exponential distribution, which has the following density function:

$$p(y) = \begin{cases} \lambda e^{-\lambda y} & y \geq 0 \\ 0 & y < 0 \end{cases}$$

(a) You have collected the duration of each student visit for several office hours. Now use this dataset $\mathcal{D} = \{y_i\}_{i=1}^n$ to estimate the parameter $\lambda > 0$ of the distribution.

i. (2 points) Write down the log-likelihood function $\ell(\lambda)$ (you don't need to simplify the equation).

Solution:

$$\ell(\lambda) = \sum_{i=1}^{n} \log \lambda e^{-\lambda y_i}$$

or

$$\ell(\lambda) = \sum_{i=1}^{n} \log p(y_i)$$

ii. (3 points) Solve for the optimal λ by maximum likelihood estimation.

Solution:

$$\lambda_{\text{MLE}} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

(b) To have a more accurate estimate for each individual student who comes to the office hour, you decide to predict the duration y given features of each student and their questions $x \in \mathbb{R}^d$ using a linear probabilistic model $p(y \mid x; w)$ where $w \in \mathbb{R}^d$ is the weight vector. You would still like to model y as a random variable from the exponential distribution, but now parametrized by $w \cdot x$ (instead of λ).

i. (2 points) Recall that we need a transfer function to map the score $w \cdot x$ to the parameter space of the exponential distribution. What is the input and output space of this map?

A. $\mathbb{R} \to [0, 1]$

B. $\mathbb{R} \to (0, 1]$

C. $\mathbb{R} \to [0, +\infty]$

D. $\mathbb{R} \to (0, +\infty]$

ii. (3 points) Based on your answer above, propose a transfer function f.

Solution:

$$f(x) = e^x$$

iii. (4 points) Give an expression for the log-likelihood of one data point (x, y): $\log p(y \mid x; w)$ using the transfer function above.

Solution: Replace λ with $e^{w \cdot x}$.

$$\log(w \cdot x) - yw \cdot x$$

2. **Bayesian methods**

(a) (4 points) Suppose we are trying to estimate a parameter θ from data D. What is the difference between a maximum likelihood estimate and a maximum a posteriori estimate of θ? Use the mathematical expressions for both quantities.

Solution: The MLE is the θ under which the observed data has the highest likelihood:

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} P(D|\theta) \tag{1}$$

whereas the MAP is the θ whose posterior probability is highest, given a prior:

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta} P(D|\theta)P(\theta) \tag{2}$$

Suppose we are trying to estimate the parameters of a biased die, that is, each trial is a draw from a categorical distribution with probabilities $(\theta_1, \ldots, \theta_6)$ (for example, $P(X = 1) = \theta_1$).

(b) (3 points) We throw the die N times. The throws are independent of each other. What is the likelihood of an outcome $(n_1, \ldots, n_6)$, where $\sum_{i=1}^{6} n_i = N$ and n_1 represents the number of trials where the result was 1?

Solution: Because the throws are independent we can multiply the likelihoods of the outcomes of each individual throw:

$$\prod_{i=1}^{6} \theta_i^{n_i} \tag{3}$$

(c) (4 points) We will use the Dirichlet prior, which has the following form, where $\alpha_1, \ldots, \alpha_6$ are parameters:

$$f(\theta_1, \ldots, \theta_6) \propto \prod_{i=1}^{6} \theta_i^{\alpha_i - 1} \tag{4}$$

Given this prior, what is the posterior, up to the normalizing constant (i.e. you can use the $\propto$ symbol in your response)?

Solution:

$$f_{\text{posterior}}(\theta_1, \ldots, \theta_6) \propto \prod_{i=1}^{6} \theta_i^{\alpha_i - 1} \prod_{i=1}^{6} \theta_i^{n_i} = \prod_{i=1}^{6} \theta_i^{\alpha_i + n_i - 1} \tag{5}$$

(d) (3 points) Is the Dirichlet prior a conjugate prior to the categorical distribution? Explain your answer.

Solution: Yes, because the posterior is in the same parametric family as the prior: they're both Dirichlet distributions of the form

$$\prod_{i=1}^{6} \theta_i^{\gamma_i - 1} \tag{6}$$

In the prior $\gamma_i = \alpha_i$ and in the posterior $\gamma_i = \alpha_i + n_i$.

3. **Dog classification App.** Alice is trying to build an App to help beginning dog lovers recognize different types of dogs. To start with an easy setting, she decides to build a classifier with three classes: Golden Retriever, Husky, Not Dog (class 1, 2, and 3).

   (a) First, she decides to use the AvA (all vs all) method, where each pairwise classifier h_{ij} is a binary classifier that separates class i (positive) from class j (negative).

      i. (3 points) Consider the following outputs from each classifier: $h_{12}(x) = 1$, $h_{13}(x) = -1$, $h_{23}(x) = -1$. What prediction should the App make for an example x? Your answer should be Golden Retriever, Husky, or Not Dog.

      Solution: Not Dog

      ii. (3 points) How many *more* classifiers does Alice need to train to update her App to include German Shepherd?

      Solution: 3

   (b) A friend who tried the initial version of the App gave the feedback that it is really annoying seeing the App sometimes classifies a dog as "Not Dog ". To improve user trust, Alice then decides to penalize the model more when it classifiers either Golden Retriever or Husky as Not Dog.

      i. (3 points) To achieve this, Alice start with the multiclass zero-one loss $\Delta(y, y')$. Recall that in class we defined $\Delta(y, y')$ to be $\mathbb{I}\{y \neq y'\}$. Now, redefine $\Delta(y, y')$ for Alice by filling the table below to indicate the preference that confusion between any dog and "Not Dog " incur a penalty of 5 whereas other misclassification incurs a penalty of 1.

$\Delta(y, y')$	y' = Golden Retriever	y' = Husky	y' = Not Dog
y = Golden Retriever			
y = Husky			
y = Not Dog			

Solution:

$\Delta(y, y')$	y' = Golden Retriever	y' = Husky	y' = Not Dog
y = Golden Retriever	0	1	5
y = Husky	1	0	5
y = Not Dog	5	5	0

ii. (6 points) Next, Alice adapts the generalized hinge loss by plugging in the new Δ function defined above:

$$\ell_{\text{hinge}}(y, x, w) \stackrel{\text{def}}{=} \max_{y' \in \mathcal{Y}} \left(\Delta(y, y') - \langle w, (\Psi(x, y) - \Psi(x, y'))\rangle\right)$$

For simplicity, let $s(x, y)$ denote the compatibility score $\langle w, \Psi(x, y)\rangle$. Given an image x, consider the following prediction:

- $s(x, \text{Golden Retriever}) = 1.5$
- $s(x, \text{Husky}) = 4.5$
- $s(x, \text{Not Dog}) = 3.5$

What is the loss with respect to different groundtruth labels y? (Your answer should be a real number)

- $\ell(x, \text{Golden Retriever}, w) =$
- $\ell(x, \text{Husky}, w) =$
- $\ell(x, \text{Not Dog}, w) =$

Solution:

- $\ell(x, \text{Golden Retriever}, w) = 7$
- $\ell(x, \text{Husky}, w) = 4$
- $\ell(x, \text{Not Dog}, w) = 6$

4. **Decision trees and ensemble methods**

For each of the following statements, indicate whether it is true or false, and explain your answer.

(a) (3 points) Decision trees favor high-variance features; centering and scaling the values of all features before fitting the tree can counteract this tendency.

Solution: False. Fitting procedures for decision trees do not take the variance of a feature into account, and monotonic transformations of feature values don't affect the outcome of these procedures.

(b) (3 points) Boosting is more difficult to parallelize than bagging.

Solution: True. Bagging involves fitting many independent models, which is easy to parallelize, but boosting is serial: model n depends on model $n-1$.

(c) (3 points) Decision trees are generally less sensitive to outliers than linear regression.

Solution: True. In a decision tree, an outlier will only affect the model's predictions for instances that fall in the region that includes the outlier; in linear regression, an outlier will affect the prediction for all instances.

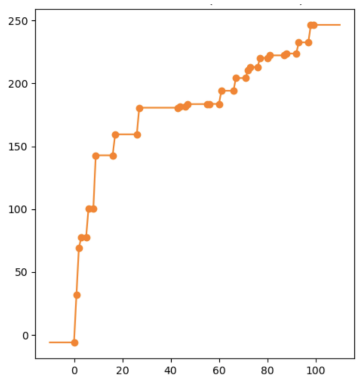
(d) (3 points) Bagging leverages the fact that the decision tree fitting procedure we discussed in class is nondeterministic: because fitting a tree to the same dataset results in a different tree each time, averaging the predictions of multiple such trees reduces variance.

Solution: False. Bagging does involve averaging the predictions of multiple trees, but the source of variability in those trees comes from bootstrapping (resampling the training data with replacement). Fitting a decision tree to a fixed dataset always results in the same decision tree.

5. **Monotonic regression.** You are going to use gradient boosting to solve a regression problem, i.e. learn a function $f : \mathbb{R} \to \mathbb{R}$ where

$$f \in \left\{ \sum_{m=1}^{M} v_m h_m(x) \mid v_m \in \mathbf{R},\ h_m \in \mathcal{H},\ m = 1, \ldots, M \right\}$$

Let's assume that the true function is monotonically increasing. For example:



(a) (2 points) What loss function $\ell(y, \hat{y})$ (where $\hat{y}$ is the prediction) will you use?

Solution: squared loss

(b) (3 points) What is the pseudo residual for an example (x_i, y_i)? (You can directly use f in the expression)

Solution:

$$-(y_i - f(x_i))$$

It's also okay to include the coefficient 2.

(c) (3 points) Does it make sense to choose the base hypothesis space $\mathcal{H}$ to be all linear predictors ($h(x) = w \cdot x$)? Why or why not?

Solution: No. Because sum of linear functions is still linear.

(d) (3 points) Which of the following base predictors ($w \in \mathbb{R}$ is the parameter) ensures that f will be monotonic? (select all that apply)

A. $h(x) = \begin{cases} 1 & x > w \\ 0 & x \leq w \end{cases}$

B. $h(x) = \begin{cases} x & x > w \\ 0 & x \leq w \end{cases}$

C. $h(x) = |x - w|$

D. $h(x) = \frac{1}{1+e^{-(x-w)}}$

6. **Neural networks**

Suppose our inputs have binary features $x_1 \in \{0, 1\}$ and $x_2 \in \{0, 1\}$, and we would like to define a classifier f where $f(x_1, x_2) = 1$ if $x_1 = 1$ or $x_2 = 1$, but not both, and $f(x_1, x_2) = 0$ otherwise. For the following questions, you can use the sign function $\sigma(x) = 1_{x \geq 0}$ as the nonlinearity, for simplicity.

(a) (2 points) Can a single-layer neural network (with no hidden units) compute this function? If so, spell out the equations computed by the network; if not, explain why.

Solution: No, because a single-layer neural network can only compute linear functions, and the decision boundary here is nonlinear; this plot was shown in lecture 12.

(b) (2 points) Can a classification tree compute this function? If so, draw the tree; if not, explain why.

Solution: Pretty obvious.

(c) (3 points) Can a two-layer neural network (with one hidden layer) compute this function? If so, spell out the equations computed by the network and provide an appropriate set of weights; if not, explain why.

Solution:

$$h_1 = \sigma(x_1 + x_2 - 1.5) \tag{7}$$
$$h_2 = \sigma(-x_1 - x_2) \tag{8}$$
$$o = -\sigma(h_1 - h_2) \tag{9}$$

In the following questions we will optimize the parameters for ridge regression using backpropagation. Recall that ridge regression is linear regression with the regularization term $\lambda w^T w$, where w are the weights and λ is a hyperparameter. We're going to use the standard square loss.

(d) (2 points) What is the objective function for a training instance (x, y)?

Solution: $(y - (w^T x + b))^2 + \lambda w^T w$ (or without b if the bias term is absorbed into the inputs).

(e) (4 points) Draw the computation graph for this function, and below the graph express each node explicitly as a function of its inputs (for example, the last node in the graph might be $J = L + R$, where J is the objective, R is the regularization term and L is the loss).

(f) (6 points) Using backpropagation, compute the partial derivative of the objective function with respect to a particular weight w_j. Show all intermediate derivatives. You will not be penalized for calculus mistakes (though try not to make them anyway!).

Solution:

$$\frac{\partial J}{\partial w_j} = \frac{\partial J}{\partial l}\frac{\partial l}{\partial r}\frac{\partial r}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial w_j} + \frac{\partial J}{\partial R}\frac{\partial R}{\partial w_j} = 1 \times 2r \times (-1) \times x_j + 1 \times 2\lambda w_j \tag{10}$$

7. **K-means clustering of a toy dataset.** Recall that the k-means algorithm aims to minimize the following objective on a dataset $\mathcal{D} = \{x_i\}_{i=1}^n$:

$$J(c, \mu) = \sum_{i=1}^{n} \|x_i - \mu_{c_i}\|^2 ,$$

where $c_i \in 1, \ldots, k$ is the cluster assignment for each example x_i, and μ_j is the centroid of cluster $j \in 1, \ldots, k$.

(a) (2 points) Suppose there is a single cluster (i.e. $k = 1$). Give an expression for the optimal centroid μ_1^* (no proof needed).

Solution:

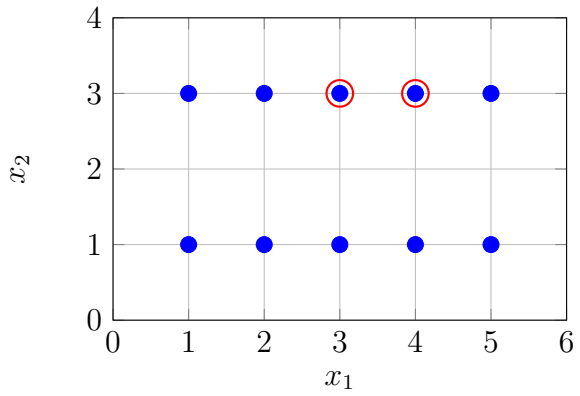
$$\mu_1^* = \frac{1}{n} \sum_{i=1}^{n} x_i$$

(b) (3 points) Suppose there is n clusters (i.e. $k = n$). Give an expression for the optimal cluster assignments c_i^* and centroids μ_i^* (no proof needed). What is the optimal objective $J(c^*, \mu^*)$ in this case?

Solution:

$$c_i^* = i$$
$$\mu_i^* = x_i$$
$$J(c^*, \mu^*) = 0$$

(c) Consider the following 2D dataset.



You will be asked to circle or mark points in the graph. You can either directly draw on the figure, or list the points by their coordinates in text.

You are going to run k-means algorithm to cluster it with $k = 2$. The points circled in red denotes the initial cluster centroids.

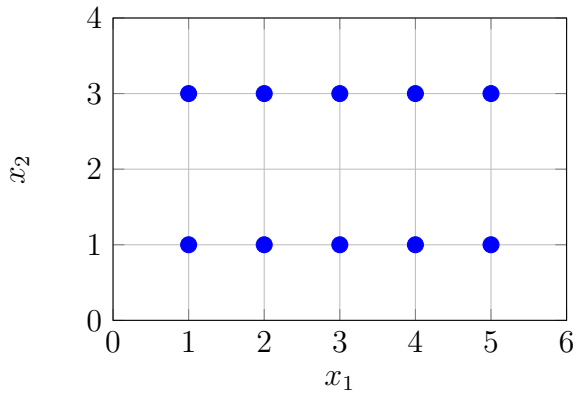
i. (3 points) Show the cluster centroids and the points in each cluster when the algorithm converges.

Solution: Left 6 in one cluster. Right 4 in the other.

ii. (3 points) Is this the clustering you would come up with by inspecting the pattern in the data? If not, indicate the two clusters that make more sense to you.

Solution: Upper 5 in one and lower 5 in the other.

(d) (4 points) Select two points (out of the ten blue dots) as initial cluster centroids such that the k-means algorithm would converge to the desired clusters that you give in the previous question.



Solution: The middle two points.

Congratulations! You have reached the end of the exam.