CS 1675 Spring 2020: Final Exam

Assigned April 20, 2020; Due: April 23, 2020

Shibo Xing

Submission time: April 23, 2020 at 9:00PM EST

Collaborators You are **not** permitted to work with anyone else for Test 02. All of the work must be your own.

Instructions

You must submit a PDF document to CourseWeb. That PDF can be created from the provided .Rmd template, or with another environment, such as Microsoft Word, LaTeX, or Google Docs. If you use the .Rmd template you are allowed to create as many equations blocks as you need, and add as much as markdown text as you need to answer the questions. If you use another environment (such as Microsoft Word), you do not need to retype the problem statements. Regardless of submission method, your solutions must be clearly marked and easy to find.

There are 7 Problems. Each problem consists of multiple sub-parts. All sub-parts are worth 2 points, **except** Problems 5d), 3c), and 3d). Problem 5d) and 3c) are worth 5 points each. Problem 3d) is worth 10 points.

Problem 01

You are interested in fitting a single layer neural network with 2 inputs, x_1 and x_2, to predict a continuous response, y. You will use a neural network with 2 hidden units, h_1 and h_2. The hidden unit parameters will use the notation β_{dk} to represent the d-th input's weight for the k-th hidden unit. The output layer parameters will be denoted as α_k to represent the k-th hidden unit's output layer weight. The neural network response will be denoted as f.

1a)

Write out the expression for the n-th observation of the first hidden unit, $h_{n,1}$, where the non-linear transformation function is to be denoted by the generic expression $g(\cdot)$. You do **not** need to use matrix notation.

$$h_{n,1} = g([1\ x_n] \cdot \beta_{:,1})$$

1b)

Write out the expression for the n-th observation of the second hidden unit, $h_{n,2}$, where the non-linear transformation function is to be denoted by the generic expression $g(\cdot)$. You do **not** need to use matrix notation.

$$h_{n,2} = g([1\ x_n] \cdot \beta_{:,2})$$

1c)

Write out the expression for the n-th observation's neural network response, f_n, in terms of the hidden unit responses and the output layer parameters.

where a0 is a column vector of the same bias value

$$f_n = a_0 + h_n \cdot a_n$$

1d)

In lecture, we discussed several options for the non-linear transformation function $g(\cdot)$. In this problem, you will consider using a linear transformation, such that the $n - th$ observation's k-th hidden unit's response, $h_{n,k}$, is equal to:

$$h_{n,k} = g(\eta_{n,k}) = \eta_{n,k}$$

where $\eta_{n,k}$ is the linear predictor for the n-th observation of the k-th hidden unit.

Derive the expression for the neural network response, f_n, as a function of the inputs and unknown parameters, assuming the linear transformation function. Thus, you must derive an expression that does not depend on either of the $h_{n,k}$ or $\eta_{n,k}$ terms.

where a0 is a column vector of the same bias value

$$f_n = \alpha_0 + [1 \ x_n] \cdot \beta \cdot \alpha$$

1e)

Why do you think it is so important to use non-linear transformation functions in neural network models?

Solution:

If we have only linear transformations, then all of the layers of unknown parameters could be calculated just with one matrix multipliation, in which case the nerual network will only have one layer of coeffcients to learn.

Problem 02

You will continue working with a single layer neural network with 2 inputs and 2 hidden units. The N training inputs are stored in a design matrix **X** which includes an additional intercept column of 1's. The hidden unit parameters are stored in column vectors $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$.

2a)

In lecture we discussed calculating all hidden unit linear predictors with matrix math by assembling the matrix of all hidden unit parameters, **B**. Write out the expression for the matrix of all hidden unit linear predictors, **A**, using the **X** and **B** matrices.

$$A = XB$$

2b)

If there are $N = 200$ training points, what is the dimensionality of the $\mathbf{B}$ matrix?

$$3 \times 2$$

2c)

What is the dimensionality of the $\mathbf{A}$ matrix?

$$N \times 2$$

2d)

The linear predictors are transformed through a non-linear transformation function, $g(\cdot)$. The non-linear hidden unit responses are stored in the $\mathbf{H}$ matrix. Write out the neural network response in matrix notation, $\mathbf{f}$, using the output layer weight column vector α and the output layer bias, α_0.

$$H = g(A)$$
$$f = H\alpha + \alpha_0$$

2e)

How many output layer parameters would a neural network model have if you used 5 hidden units, instead of the 2 hidden units? Continue to assume that there are 2 inputs.

solution: α vector will have 5 parameters α_0 will still be a single unit. together 6 parameters.

2f)

What is the dimensionality of the $\mathbf{B}$ matrix if we use 7 inputs instead of 2 inputs? Assume that we are using the model with 5 hidden units.

$$8 \times 5$$

Problem 03

In lecture and in the homework assignments, we mostly focused on neural network models for regression. However, we discussed that neural networks can be used for classification tasks by including an additional non-linear transformation function to the output layer. In this problem you will specifically consider a binary classification task. The binary outcome, y_n, equals 1 to represent the event occurred and it equals 0 if the event did not occur. If the n-th observation's continuous output layer response is denoted as f_n, the transformed output for the event probability, μ_n, is calculated by passing f_n through the logistic function:

$$\mu_n = \text{logit}^{-1}(f_n)$$

The continuous response, f_n, depends on the output layer parameters, α_k, and the hidden unit responses, $h_{n,k}$. You will assume that there are 3 hidden units and 5 inputs for this problem.

3a)

Write out the expression for the n-th observation's continuous response in terms of the hidden unit responses and the output layer parameters. You do **not** need to use matrix notation.

$$f_n = \alpha_0 + \sum_{k=1}^{3}(h_{n,k} \cdot \alpha_k)$$

3b)

Binary classification problems require a different loss function compared to regression problems. The loss function to minimize for binary classification is the binary cross entropy loss function. The n-th observation's contribution to the loss function will be denoted as L_n. The "complete" loss function requires summing over all N observations:

$$L = \sum_{n=1}^{N}(L_n)$$

Write out the expression for the n-th observation's binary cross entropy loss, L_n, in terms of the observed binary outcome y_n and the neural network event probability μ_n.

$$L_n = -y_n \cdot log(\mu_n) - (1 - y_n) \cdot log(1 - \mu_n)$$

3c)

The backpropagation equations allow calculating the gradient of the loss with respect to all parameters. We derived the backpropagation equations for a regression task in lecture. However, the output layer logistic function must be accounted for when deriving the backpropagation equations for the neural network binary classifier. Although there are two sets of parameters, the output layer parameters α_k and the hidden layer parameters β_{dk}, you will only work with the output layer parameters in this problem.

Before deriving the partial derivative of L_n with respect to the output layer parameters, α_k, let's break up the partial derivative using the chain rule. Specifically, you must write the partial derivative of L_n with respect to the first hidden unit's output layer weight, α_1, with the chain rule.

You do not need to complete the derivation in this question. You must write out the partial derivative:

$$\frac{\partial L_n}{\partial \alpha_1}$$

in terms of other "component" derivatives.

$$\frac{\partial L_n}{\partial \alpha_1} = \frac{\partial(y_n \cdot log(\mu_n))}{\partial \alpha_1} + \frac{\partial((1 - y_n) \cdot log(1 - \mu_n))}{\partial \alpha_1}$$

$$= \frac{\partial(y_n \cdot log(\mu_n))}{\partial \mu_n} \cdot \frac{\partial(logit^{-1}(f_n))}{\partial f_n} \cdot \frac{\partial f_n}{\partial \alpha_1} - \frac{\partial((1 - y_n) \cdot log(1 - \mu_n))}{\partial \mu_n} \cdot \frac{\partial(logit^{-1}(f_n))}{\partial f_n} \cdot \frac{\partial f_n}{\partial \alpha_1}$$

3d)

Derive the expression the partial derivative of L_n with respect to α_1.

HINT: A certain simple model could help…

$$\frac{\partial(y_n \cdot log(\mu_n))}{\partial\mu_n} \cdot \frac{\partial(logit^{-1}(f_n))}{\partial f_n} \cdot \frac{\partial f_n}{\partial\alpha_1} - \frac{\partial((1-y_n)\cdot log(1-\mu_n))}{\partial\mu_n} \cdot \frac{\partial(logit^{-1}(f_n))}{\partial f_n} \cdot \frac{\partial f_n}{\partial\alpha_1}$$

$$= \frac{y_n}{\mu_n} \cdot \frac{e^{f_n}}{(1+e^{f_n})^2} \cdot h_{n,1} - \frac{1-y_n}{1-\mu_n} \cdot \frac{e^{f_n}}{(1+e^{f_n})^2} \cdot h_{n,1}$$

3e)

You worked through the partial derivative of L_n, where the loss is the binary cross-entropy. For most of the semester however, we focused on maximizing likelihoods (or posteriors), rather than minimizing loss functions.

What is the analogous likelihood function that is maximized when the binary cross-entropy loss function is minimized?

solution: Bernoulli likelihood

Problem 04

If you are using the Rmarkdown template to answer the questions in the exam, type your answer to the multiple choice questions next to the **Your Answer:** line below the choices.

4a)

Recursive binary partitioning decision trees divide an input space into non-overlapping regions. Dividing the space into binary regions is known as "splitting". Consider a regression problem where the primary performance metric is the sum of squared errors (SSE). At each split, the splitting variable and its associated value are selected such that:

a) The variance within each region is maximized

b) The SSE is minimized

c) The total within sum of squares of each region is minimized

d) None of the above

Your Answer: b

4b)

We discussed in class how a decision tree is susceptible to overfitting. Which of the following is an approach to reduce the number of terminal nodes in a decision tree in order to reduce the complexity?

a) Boosting

b) Subsampling

c) Linking

d) None of the above

Your Answer: d

4c)

Bagging is an approach which uses resampling to generate multiple realizations of the training set. Which type of resampling procedure is used in Bagging?

a) Bootstrap

b) Regularization

c) Cross-validation

d) Clustering

Your Answer: a

4d)

Should we be worried that increasing the number of trees in a bagged tree model will lead to overfitting? Why or why not?

solution: We shouldn't be worried. Bagging method simply lets us use the average of the predictions from all the trees as the answer, but the number of times each model is trained doesn't increase as the number of trees increases.

4e)

The random forest is an extension of the bagged tree model. The primary tuning parameter is `mtry`, the number of randomly selected variables to consider at each split. Why would a low value of `mtry` help when there are many correlated input variables, compared to a standard bagged tree model?

solution: Since there are many paramters in our case, we have to retain some correlation among the trees. If we have a high mtry then we have a higher chance to miss out the strong predictor in the splits, in which case the average response might not be sensible.

4f)

Consider using a random forest binary classifier. There are 25 inputs. What value of `mtry` would turn the random forest model into a standard bagged tree model?

a) 5

b) 12

c) 25

d) 0

Your Answer: c

4g)

Boosting is an ensemble method which sequentially improves a weak learner. A boosted tree model for regression fits each tree iteratively to what quantity from the previous tree?

a) The variance of the previous tree

b) The error from the previous tree

c) The learning rate from the tree

d) The dendogram from the previous tree

Your Answer: b

4h)

You fit a boosted tree model with a very low learning rate. You anticipate this model to require how many trees relative to a boosted tree model with a very high learning rate?

a) More trees

b) Less trees

c) Equal number of trees

d) Depends on the weight decay

Your Answer: a

Problem 05

In lecture we discussed how we can use Singular Value Decomposition (SVD) to perform Principal Components Analysis (PCA). The SVD of a matrix $\mathbf{X}$ is:

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

5a)

Which of the three matrices, $\mathbf{U}$, $\mathbf{S}$, or $\mathbf{V}$ are related to the eigenvalues of the covariance matrix of the variables contained in the original matrix $\mathbf{X}$?

solution: S

5b)

Which of the matrices correspond to the eigenvectors of the covariance matrix of **X**?

solution: V

5c)

TRUE or FALSE: The first principal component (PC) is uncorrelated to the second PC, but is correlated to all other PCs.

Your Answer: FALSE

5d)

What is the reconstruction error equal to if all PCs are used to reconstruct the original variables?

solution: 0

5e)

In lecture we discussed several approaches to choosing the number of PCs to retain. Which of the following is **not** an approach?

a) Visual "knee or elbow bend" from a scree plot

b) Proportion of variance explained

c) Eigenvalue criterion

d) Hartigan's rule

Your Answer: d

5f)

Why is it usually a good idea to standardize the variables before performing PCA?

solution: A high standard error of a variable means a high weight of that variable. We can normalize the variables to let them have the same weight.

5g)

Suppose we are interested in clustering the observations of a data set together. Whether we decide to use Kmeans or Hierarchical clustering, similarity between observations is calculated based on what?

a) Variable variances

b) The Likelihood

c) Distance

d) Variable averages

Your Answer: c

Problem 06

In this problem you will consider a situation consisting of N observations of D inputs and a continuous response y. The inputs are assembled into a design matrix, $\mathbf{X}$, which includes a column for the intercept term in a linear model. The vector of all N responses is $\mathbf{y}$.

6a)

Our linear model's likelihood function is a Gaussian distribution. In addition to the unknown $\boldsymbol{\beta}$ parameters of the linear model, what other parameter is unknown?

a) The likelihood noise, σ

b) The probability of the event, μ

c) The logistic function

d) The $\boldsymbol{\beta}$ parameters are the only unknowns

Your Answer: a

6b)

Learning the unknown $\boldsymbol{\beta}$ parameters by maximizing the likelihood function is equivalent to finding their estimate which minimizes what?

a) The cross-entropy

b) The variance of the $\boldsymbol{\beta}$ parameters

c) The marginal likelihood

d) The sum of squared errors

Your Answer: d

6c)

Which of the following expressions corresponds to the Maximum Likelihood Estimate (MLE) for the $\boldsymbol{\beta}$ parameters?

a) $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$

b) $\sigma\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$

c) $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$

d) The MLE does not have a closed form solution

Your Answer: a

6d)

If we would use a Bayesian linear model, what else must be specified to complete the model setup?

a) The σ value

b) How we will remove outliers

c) The prior distributions for all unknown parameters

d) The marginal likelihood

Your Answer: c

6e)

We will use the Laplace Approximation to fit our Bayesian linear model. Which of the following is **not** a step of the Laplace Approximation?

a) Find the posterior mode through optimization

b) Determine the Hessian matrix at the posterior mode

c) Calculate R-squared at the posterior mode

d) Invert the Hessian matrix

Your Answer: c

6f)

The Laplace approximation approximates the posterior distribution in what way?

a) The kernel trick

b) A Multinomial

c) A Multivariate Normal

d) Independent Gaussians

Your Answer: c

Problem 07

7a)

Lasso regression is analogous to a Bayesian linear model with what kind of prior distribution?

solution: Laplace prior distributions

7b)

Bayesian inference for a constant unknown event probability μ with a Binomial likelihood and a Beta prior distribution results in what type of posterior distribution on μ?

solution: Beta distribution

7c)

What makes a linear model linear?

solution: the fact that the coefficiencts (unknown parameters) have a linear relationship

7d)

TRUE or FALSE: Logistic regression does not have a closed form solution for the Maximum Likelihood Estimates (MLEs) of the unknown parameters.

Your Answer: TRUE

7e)

Fitting logistic regression models by maximizing the liklihood breaks down under what condition?

solution: linearly separated classes, a diffuse prior

7f)

The basic definition of probability is: the likelihood of an event