CS 181 Spring 2020 Midterm 1 Review

1 Purpose

The purpose of this session is to provide a summary of the material we have covered in the course thus far. In these notes we present high-level summaries of the important concepts in the course, along with important equations and techniques you should know. This is not a substitute for your own independent studying.

2 Regression

2.1 Linear Regression

2.1.1 Summary

We may want to predict one variable as a function of another variable. For example, we can predict height based on weight. We write the loss as

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \mathbf{w}^\top \mathbf{x}_i\right)^2 \tag{1}$$

If we minimize the function with respect to the weights, we get the following solution:

$$\mathbf{w}^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} = \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) \tag{2}$$

2.1.2 Concept Questions/Important Skills

- What is the bias trick?
- Why is there a squared term in the loss?
- You should be familiar with the skills necessary to derive the solution for w^* because you may be asked to solve similar problems on the midterm. Sketch the steps of how to derive w^*.
- What is a basis function, and why would we use it?

2.2 Regularization

2.2.1 Summary

Sometimes if we allow our model to have too many parameters over a small training set, it may overfit the data. This means the model is learning the data points, not the relationship between variables. In order to prevent this, we use regularization.

Recall that the standard linear regression problem, known as *ordinary least squares (OLS)*, uses the following loss function (which is actually the mean squared error):

$$\mathcal{L}_{OLS}(D) = MSE = \sum_{i=1}^{n}(y_i - h(x_i; \mathbf{w}))^2$$

Regularization refers to the general practice of modifying the model-fitting process to avoid overfitting and other potential problems like multicollinearity. Linear models are typically regularized by adding a *penalization term* to the loss function. The penalization term is simply any function p of the weights $\mathbf{w}$ scaled by a penalization factor λ. The loss then becomes:

$$\mathcal{L}_{reg}(D) = \sum_{i=1}^{n}(y_i - h(x_i; \mathbf{w}))^2 + \lambda p(\mathbf{w})$$

There are some common choices for $p(\mathbf{w})$ that will be discussed. They frequently leverage the idea of a vector norm, where $||\mathbf{w}||_n$ represents the L_n-norm of the vector $\mathbf{w}$ for $n \geq 1$:

$$||\mathbf{w}||_n = \left(\sum_{i=1}^{|\mathbf{w}|} |\mathbf{w}_i|^p\right)^{1/p}$$

If the L_1 norm is used, we call this LASSO regression. If the L_2 norm is used, we call this ridge regression. If a linear combination of L_1, L_2 norms are used, we call this elastic net. Although we cannot solve LASSO analytically, we can solve Ridge analytically to get

$$\mathbf{w}_{ridge} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

2.2.2 Concept Questions/Important Skills

- What is sparsity, and how does it relate to regularization?
- Given a model that has much higher test than train accuracy and a model that has much higher train than test accuracy, which would you regularize (or both)?
- Why does LASSO not have an analytic solution?
- You should be familiar with the mathematical techniques required to find $\mathbf{w}_{ridge}$. Sketch the derivation, and how is this similar to the derivation for w^* for OLS?
- What is the bias variance tradeoff, and how does it relate to overfitting, underfitting, regularization, and increasing the size of the training set?

2.3 Probabilistic Linear Regression

2.3.1 Summary

Some of the choices that we made when doing linear regression may seem a bit arbitrary. For example, why use ^{2} in the loss term instead of ^{4}? In our study of probabilistic linear regression, we saw another way to derive the equations we already found, giving another way to interpret them.

We begin with a generative process, which is like a "story" for how the data were created (think of this as analagous to the story of a distribution from Stat 110). In particular, suppose that for each data point, the way y_n is generated is given by the distribution (recalling our height, weight example, this could be the story for how people's heights are determined based on weight, genes, and noise)

$$y_n \sim \mathcal{N}(w^T x_n, \beta^{-1})$$

We then create a likelihood function $p(y|X, w, \beta)$, and we optimize this with respect to w. This is called Maximum Likelihood Estimation (MLE) because we are maximizing the likelihood function. This will yield the same result as OLS.

2.3.2 Concept Questions/Important Skills

- What are the two interpretations of OLS based on this derivation and the derivation presented in 2.1.1?
- You should have enough mathematical familiarity to demonstrate how this setup and the setup in 2.1.1 are the same. Sketch the steps you would take to do this.
- How do we use logs in this scenario?

2.4 Bayesian Linear Regression

2.4.1 Summary

We can also approach the problem of linear regression from a Bayesian perspective. In the past, we've calculated point estimates of our parameters w, but we may want to capture the entire distribution $p(w|D)$ (one reason for this could be wanting to know how certain we are of our w estimate, which could be important in medical applications).

Again, we have a generative process (story) for the data. In particular, we may think that given some parameter w, this story means there's a certain probability that we see data D, so we have some $p(D|w)$. We also place a prior $p(w)$ (which represents our beliefs before seeing any data of what w is). Given the generative model, we can find the **posterior** distribution for θ

$$p(\theta|D) \propto p(D|\theta)p(\theta) \tag{3}$$

Finding the θ that maximizes this is called the Maximum A Posteriori (MAP) estimate. It is called such because you are finding the θ that maximizes the posterior distribution.

2.4.2 Concept Questions/Important Skills

- Why can we maximize the proportionality in 3? Why does this proportionality hold?
- How does bayesian linear regression relate priors and regularization?
- You should be able to do MAP estimation.

3 Classification

3.1 Binary Linear Classification

3.1.1 Summary

Suppose we have data points and we want to classify them into two classes. For example, we may want to classify people into likely or unlikely to get a certain disease so we can do early monitoring. For a data point x, we predict $\hat{y}$ with the discriminant function

$$\hat{y} = \text{sign}(h(\mathbf{x}; \mathbf{w}, w_0)) = \text{sign}(\mathbf{w}^\top \mathbf{x} + w_0)$$

We use the loss function (note $ReLU(z) = \max\{0, z\}$)

$$\begin{aligned}\mathcal{L}(\mathbf{w}) &= \sum_{i=1}^{n} ReLU(-h(\mathbf{x}_i; \mathbf{w}, w_0)y_i) \\ &= -\sum_{i=1:y_i \neq \hat{y}_i}^{n} (\mathbf{w}^\top \mathbf{x}_i + w_0)y_i\end{aligned}$$

- Two classes divided by a linear separator in feature space.
- **Discriminant function** : Function that directly assigns each vector to a specific class
- $\mathbf{w}$ is orthogonal to every point on the decision surface. It determines orientation of decision boundary.

We can then numerically approximate the optimal w via gradient descent

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \frac{\partial}{\partial \mathbf{w}} \mathcal{L}^{(i)}(\mathbf{w}) = \mathbf{w}^{(t)} + \eta y_i \mathbf{x}_i,$$

Perceptron refers to doing this algorithm, but only calculating the loss over one point. This is an example of a technique called stochastic gradient descent (doing gradient descent but at each step only calculating the gradient over a random sample of the points).

3.1.2 Concept Questions/Important Skills

- Why do we need to use ReLU? Why can't we use $0/1$ loss?
- Why do we have to use gradient descent?
- What is the idea behind gradient descent (ie intuitively, why does it work)? What happens if the learning rate is too high? Too low?
- What are some benefits and harms to stochastic gradient descent?
- If the data is linearly separable, are there any guarantees about perceptron's behavior? What is the intuition behind this?

3.2 Logistic Regression

3.2.1 Summary

Binary linear classification gives us predictions, but we may want a model that captures a higher level of granularity. For example, you may want your model to be able to distinguish between a person who has a 0.6 chance of being sick and a person who has a 0.95 chance of being sick. Our first idea will be to turn to begin with a linear model $\mathbf{W}^T\mathbf{x} = \mathbf{z}$ (predicting a value $\mathbf{w_i}^T\mathbf{x}$ for each class), but this does not give us a vector of probabilities. We use a function called softmax to get a vector of probabilities

$$\text{softmax}_k(\mathbf{z}) = \frac{\exp(z_k)}{\sum_{i=1}^{K} \exp(z_i)} \tag{4}$$

We then use then maximize the likelihood

$$p(\{\mathbf{x_i}, \mathbf{y_i}\}_{i=1}^{N} | \mathbf{W}) = \prod_{i=1}^{N} \prod_{j=1}^{K} p(\mathbf{y_i} = C_k | \mathbf{x_i})^{y_{ij}} \tag{5}$$

3.2.2 Concept Questions/Important Skills

- Verify that softmax actually gives a vector of probabilities.
- What can we say about the shape of the decision boundary of a logistic regression?
- Why is there an y_{ij} in the exponent? Can you explain all the parts of the likelihood function
- How do we find an optimal $\mathbf{W}$? What techinques do we use?
- How is this different for just two classes? Compare your answer to the week 2 section notes for comparison.
- You should be able to do MLEs to solve this type of problem.

3.3 Probabilistic Generative Classification

3.3.1 Summary

In logistic regression, our goal was to model $p(y|x)$ so that we could assign data points to classes. However, we might think it's a better idea to try and model the entire process (ie story). In this case, we assume a generative model and try to model the distribution $p(x, y)$. This factors into $p(x|y)p(y)$.

- $p(y)$ is called the **class prior** and is always a categorical distribution.
  Gives an a priori probability of an observation being a certain class, without even considering the observation's features.
- $p(x|y)$ is called the **class-conditional distribution** and its form is model-specific.
  Specifies how likely an observation (set of features) is given a class.

We are interested in picking the class k that maximizes $p(y = k|\mathbf{x})$.

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} \propto p(\mathbf{x}|y)p(y) \tag{6}$$

3.3.2 Concept Questions/Important Skills

- Why can we maximize the proportionality in 6? How do we know this is a proportionality?
- You should be comfortable deriving the optimal parameters for a generative model. Describe the steps you would take to do this MLE estimate? In general, have the MLE estimates been intuitive?
- What is the Naive Bayes assumption, and how would we use it in this context?

4 Neural Networks

4.0.1 Summary

As we've discussed, we often need basis transformations in order to make sure our models are able to accurately capture the relationships in our data. Neural Networks are an extremely flexible model that allow us to fit complex problems without such feature engineering.

A neural network consists of nodes in layers. Let our data be composed of pairs (x, y). Each node in the first layer (input layer) is a feature of the data. Similarly, each node in the output layer corresponds to a dimension of our output y. The layers in the middle are called hidden layers, and are composed of an activation function (such as ReLU or sigmoid) applied to a linear combination of the outputs from the previous layer's nodes. We will then train the network according to a loss, which may differ based on application (ie if the network is being used for a regression or classification problem).

4.0.2 Concept Questions/Import Skills

- Why do we need activation functions?
- What is backpropogation, and why is it useful?

5 Test Taking/Preparation Tips

- Look at all the questions. If you get stuck on something, try other problems, and come back.
- Make sure to manage your time well! Budget time for each problem because the exam may likely take longer than you expect. You don't want to miss a question you would have known the answer to because you didn't have time to look at it.
- When in doubt for an MLE, guess something intuitive.
- Check that your results make sense. This is a good way to catch small mathematical errors.
- Add some of the critical matrix cookbook formulas to your cheat sheet!
- If you're pressed for time, you can use summations instead of matrix notation
- Doing the problems from the section notes is a great way to prepare for the exam!
- Study by making your own cheat sheet. It will help you understand the key concepts!