CS57300 Study Guide

1. Controlling the Model Space
    a. To avoid model overfitting, models like logistic regression and neural networks, we often include parameter priors. Under which conditions would you consider using a strong or a weak prior.
    b. Describe why adding more data decreases the chances of overfitting
        i. A good answer here will talk about hypotheses. Say, why does increasing the training data decreases the chances of have a wrong hypothesis (a troll parameter) be declared "good"?
        ii. Think of our class on hypothesis testing and the relationship between the number of observations and the likelihood a false hypothesis is declared true.
2. Decision trees. largest information gain
    a. Know how to build decision trees (DTs)
    b. Why our DT optimization is greedy, i.e., why is it impractical to do an exhaustive search over the model space?
    c. Why binary do attributes only need to be considered once in the recursive decision tree call? Why do continuous-valued attributes must be considered at every split?
    d. Is it easy for a decision tree to overfit the training data? Why? pruning

3. Naïve Bayes & Logistic Regression (LR)
    a. Explain why LR is considered a linear classifier
        i. Understand through equations
    b. Naïve Bayes
        i. What is the Naïve Bayes main assumption? conditional independent
        ii. What happens if the assumption is violated?
        iii. Describe the algorithm
4. Ensemble Methods
    a. When and why should bagging be used? Give one condition where it would it have little to no effect.
    b. Describe how Random Forests (RFs) work
        i. Why for small datasets or for datasets with lots of features, RFs tend to work better than a single decision tree?
    c. When and why should boosting be used?
        i. Describe the Adaboost algorithm
        ii. Describe a condition where it will fail
5. Give a high-level description how the additive trees of Gradient Boosted Decision Trees work
6. Neural Networks
    a. Why does a fully connected neural network layer is guaranteed to outperform a not-fully-connected layer if we have enough training data. Give an example and explain why. non-fully \in fully
    b. Why do we need nonlinear neuron activations?

        i. What would happen to a deep neural network if the activation were a linear function? Can you describe it mathematically?
    c. Describe the forward and backward passes used to train neural networks
        i. Describe how gradients flow in and out of a network layer. Which gradients a neural network layer is expecting to receive, which gradients does it propagates back to the lower layers, and how does it update its own parameters? Pay attention to clearly index the gradients using the training data.
        ii. In a ReLU unit, under which conditions can the gradients be zero?

7. Model Validation
    a. Learning curves (training and validation accuracies as training data increases)
        i. What do you expect to see?
        ii. Identify problems with model search (optimization) and whether or not we have enough data (or the model is too simple)
        iii. Peculiarities of neural network models (see homework 3)
    b. ROC curves and AUC score
    c. K-fold cross validation
    d. Bootstrapping
    e. Hypothesis testing
        i. Formalizing the hypotheses (null and alternative)
            1. One-tailed and two-tailed tests
        ii. Testing multiple hypotheses
            1. What are the issues one faces when testing multiple hypotheses
            2. Nested hypothesis tests (as in the homework)
        iii. Paired t-tests

8. Neural networks for structured data
    a. Why not use a feedforward network for images?
    b. What are the issues when trying to apply neural networks to graphs
        i. Understand the importance of graph isomorphism in this problem
        ii. Understand what constitutes an operation that is invariant to isomorphism (sum, average, max, min)

9. Multi-armed Bandits
    a. Play-the-winner (issues)
    b. Epsilon-greedy (issues with multiple arms)
    c. UCB1
        i. describe the algorithm and what it is trying to accomplish.
        ii. Be mindful of the meaning behind simple questions like, "why not use the lower bound rather than the upper bound".
        iii. What happens when the arms have similar or very different rewards?

10. Dimensionality reduction
    a. Problems with PCA's requirement of orthogonality
    b. Describe how we can use a neural network for dimensionality reduction