CS 145 Notes

Lecture Notes

<u>**10/1 - Week 1**</u>

- Lots of growth of data
- To collect data and make it available, we have automated tools, database systems, etc
- Mining in particular is the knowledge discovery from data and the extraction of interesting patterns of knowledge from huge amounts of data.
- Go from a ridiculous amount of information to some sort of actual insight that is useful.
- Traditionally, mining was done through database systems. This is the view from database perspective.
  - First store all the data that you have in the databases
  - Then data cleaning
  - Then store in data warehouse
  - Then figure out the task relevant data
  - Then look for patterns in that data
- This is the view from the business side of things.
  - Figure out data sources
  - Do data preprocessing
  - Data exploration to figure out some clues or patterns
  - Then data mining, which refers to information discovery
  - Then data presentation
- This is the view from the ML and Stats POVs
  - Take input data
  - Do data preprocessing, which involves data normalization, feature selection, dimensionality reduction.
  - Then data mining, which is pattern discovery, classification, correlation, clustering, etc.
  - Then post processing which is just evaluation and interpretation of the patterns discovered.
- In data mining, need to consider:
  - Kinds of data that can be mined
  - Patterns that can be found
  - Tech that can be used
  - _ that can be targeted
- Types of Data
  - Vector Data
    - Basically all the features are in some N dimensional vector
  - Set Data
    - Each example contains a set of objects

        - We can represent this set as a vector as well by just one hot encoding all the items in the set, but the problem is that you'll have a sparse matrix.
    - Text Data
        - Topic modeling is algorithm that takes in a classification of news articles or documents and have the output X number of major topics for each article.
        - Word embeddings to represent words in high dimensional space.
            - Word2Vec for example
    - Sequence Data
        - Genomic data is probably the most applicable
        - Everything in the sequence matters and the order is especially important.
    - Time Series Data
        - Each data point is associated with a time stamp and some associated value at that particular time.
    - Graph/Network Data
        - Shows relationships and interactions between data points through edges.
        - Cannot assume independence of all the examples in your set.
        - Allows you to run algorithms to see who is connected to who
    - Image Data
        - Each data point is an image, described by pixels.
- Types of Data Mining Functions
    - Association and Correlation Analysis
        - Frequency patterns
            - Which items are frequently purchased together
        - Association rules
            - Diaper -> Beer
            - Have to be careful with correlation and causation problem
    - Classification
        - Use training data to describe classes and concepts, and then predict some unknown class labels.
        - Methods like Log reg, decision trees, SVMs, and NNs
    - Clustering
        - Form of unsupervised learning to find distribution patterns
        - Methods like KNN
    - Some other functions are prediction, similarity search, ranking, and outlier detection

10/3 - Week 1

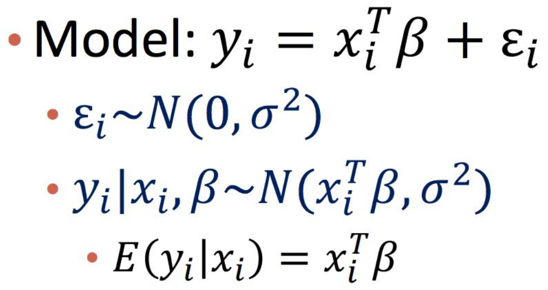
- When thinking about the algorithms, you need to think of
    - When it is applicable?
        - Type of inputs and outputs you're dealing with
        - The strengths and weaknesses of those particular algoirhtms.

    - How does it work?
        - Need to understand the pseudo-code, the workflows, and the major steps.
    - Why does it work?
        - Develop some theoretical understanding for why it is solving the problem it's solving.
- Training examples are shown using a matrix. Each row in the matrix represents a data point/example and the columns show the different attributes.
    - Dimensionality is going to be n x p where n is the number of examples and p is the number of attributes or dimensions.
- Attributes can be categorical (race, sex) or numerical (height, income)
- Categorical Attribute Types
    - Nominal: Categories or states or names of things. Basically for this attribute, you can take a bunch of discrete values
        - Hair color (Black, Brown, etc)
    - Binary: Nominal attribute with only 2 states (0, 1)
        - Symmetric binary: Both outcomes equally important
            - Gender, Sex
        - Asymmetric binary: Outcomes not equally important
            - Medical test
    - Ordinal: Values have a meaningful order, but magnitude between values isn't really known.
        - Grades (A, B, C, etc)
- Basic statistical descriptions of data
    - Central Tendency
        - Mean, Weighted arithmetic mean, trimmed mean
        - Median
        - Mode
        - Can look at symmetric vs skewed data depending on where the median, mean, and mode of the curve are located
            - If the mean is greater than the median, then it is positively skewed.
            - If the mean is less than the median, then it is negatively skewed.
    - Dispersion of the data
        - Can show quartiles, outliers, and boxplots. Q1 is the 25th percentile, Q3 is the 75th percentile
            - IQR = Q3 - Q1
            - 5 number summary: Min, Q1, median, Q3, max
            - Outlier is a value higher/lower than 1.5 x IQR
        - Variance
            - Computes a measure of scatteredness
        - Standard deviation
            - Square root of the variance

    - Histogram Analysis shows the cases that fall into particular buckets. Different from bar graph in that the area of the bar matters since the buckets can be of different sizes.
    - Scatter plot shows relationship between two variables.
        - You can be able to see whether the correlations are positive or negative or uncorrelated.
- Linear regression
    - We go from attribute values for each data point some continuous value prediction of y.
- Linear regression formalization
    - Let's say we have n data points. We have the label and the feature vectors for each of the data points. Then you do a weighted summation with some weight vector and that is your prediction.
- In this process of setting up a linreg model, you need to have
    - Model construction: Use training data to find the best parameters
    - Model selection: Use validation data to select the best model, select the best features, etc
    - Model usage: Apply the trained model to the unseen test data.
- For the estimation of the beta parameter, we need to use the cost function.
- Ordinary Least Squares regression
    - There is a closed form solution for beta. However, it's pretty expensive to be able to compute it.
    - Instead of directly solving for the closed form, performing gradient descent is another way of iteratively getting to lower cost values and in turn better parameters values.
- The difference between OLS and linear regression seems to be just in the way that you get the optimal value for the parameters.

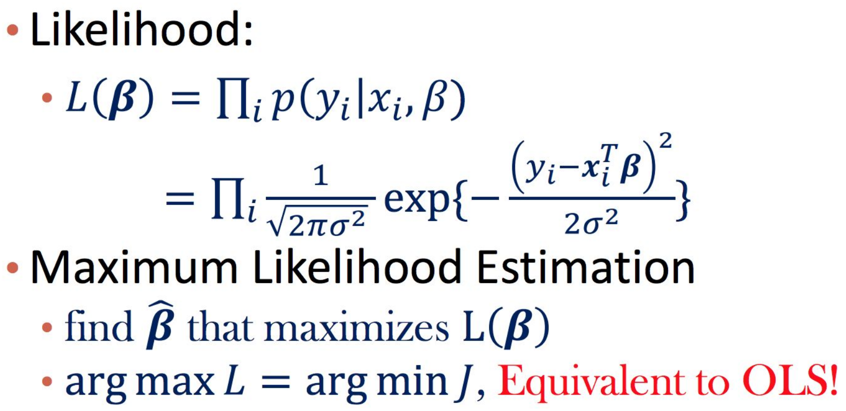
10/8 - Week 2

- Can look at the model y = x transpose beta through a probabilistic viewpoint.



- Model: $y_i = x_i^T \beta + \varepsilon_i$
    - $\varepsilon_i \sim N(0, \sigma^2)$
    - $y_i | x_i, \beta \sim N(x_i^T \beta, \sigma^2)$
        - $E(y_i | x_i) = x_i^T \beta$

- As shown above, we assume that y_i is a normal distribution with the mean of x_i transpose beta.

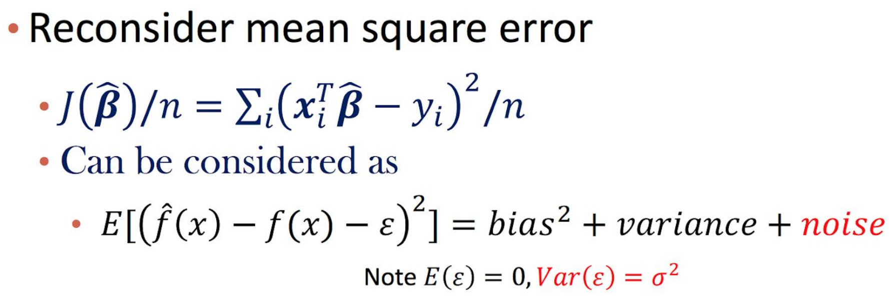


- Likelihood:
  - $L(\boldsymbol{\beta}) = \prod_i p(y_i|x_i, \beta)$

$$= \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{\left(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta}\right)^2}{2\sigma^2}\}$$

- Maximum Likelihood Estimation
  - find $\hat{\boldsymbol{\beta}}$ that maximizes $\mathrm{L}(\boldsymbol{\beta})$
  - $\arg\max L = \arg\min J$, Equivalent to OLS!

- As shown above, this is a way of looking at how to solve for beta. Instead of solving with the closed form solution or through gradient descent, we solve for the likelihood of beta.
  - The maximization of the likelihood is the same as the minimization of the cost function.
- Handling different scales of numeric attributes requires maybe using z-score.
- You can also set dummy one-hot vectors for nominal variables.
- For ordinal variables, you can just replace the values with their rank.
- Also, if X transpose X is not invertible (when you're trying to use the closed form solution), then you can add a portion of identity matrix.
- Linear Regression and OLS assumes a linear relationship between the features and the output. However, if there's no linear relationship, we could try to transform the features into x^2 or x^3, etc.
- The actual bias and variance definitions.
  - Bias asks how far is the expectation of the estimation to the true value?
    - E(f'(x)) - f(x)
    - f(x) is the groundtruth which is equivalent to x transpose beta and E(f'(x)) is the expectation of the estimator which is equal to x transpose beta prime (the learned value of beta)
  - Variance is the measure of how variant the estimator is
    - E[(f'(x) - E(f'(x)))^2]
    - If f function is very complicated, then this value will change a lot when the dataset used is changed.
  - TODO understand probabilistic interpretation of bias and variance.
- The cost function can be broken down into components that reflect the bias and variance measures of the model.



- Reconsider mean square error
  - $J(\hat{\boldsymbol{\beta}})/n = \sum_i (\boldsymbol{x}_i^T \hat{\boldsymbol{\beta}} - y_i)^2 / n$
  - Can be considered as
    - $E[(\hat{f}(x) - f(x) - \varepsilon)^2] = bias^2 + variance + noise$

Note $E(\varepsilon) = 0, Var(\varepsilon) = \sigma^2$

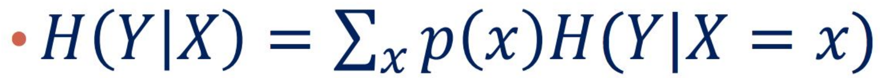
10/10 - Week 2

- We can try to find the beta that maximizes the likelihood
- Cross entropy loss is way to compute difference between two distributions.

10/15 - Week 3

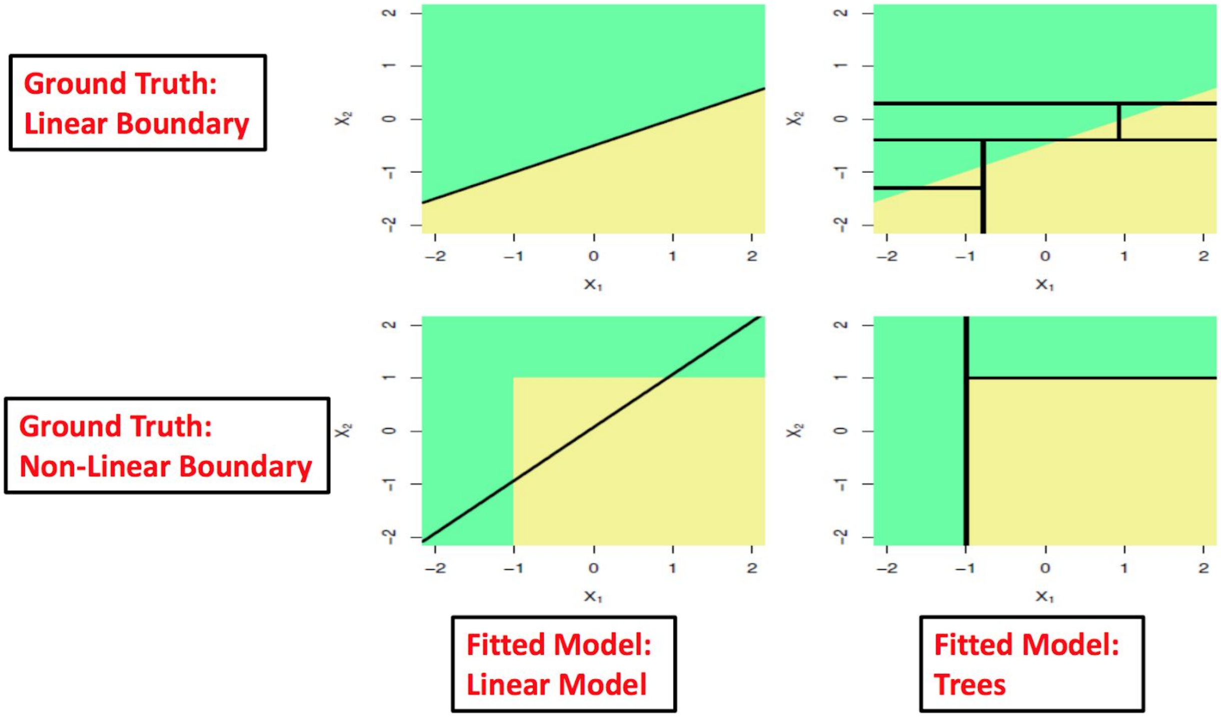
- For decision trees, you want to top-most splitting attribute to be one that, after the branching, the different classes are separated from each other.
- Entropy evaluates how much information a distribution carries.
  - Specifically, it is a measure of uncertainty associated with a random variable.
  - Higher entropy = higher uncertainty

$$H(Y) = -\sum_{i=1}^{m} p_i \log(p_i)\text{ , where } p_i = P(Y = y_i)$$

- If you have a binary class situation, and if both classes have a probability of 0.5, then you have the maximum entropy, H(X) = 1.
  - The reasoning is that if both probabilities are 0.5, then we are very uncertain about the result. Higher uncertainty means higher entropy.
- Entropy is just a measure of uncertainty based on the current distribution of data (the class labels), there aren't any predictions or any sort of model yet.
- Conditional entropy is when we calculate entropy of Y, given that we know a certain attribute X.
  - 

    $H(Y|X) = \sum_x p(x) H(Y|X = x)$
  - So, instead of calculating H(Y) from the class labels, we're asking to calculate H(Y | X) which is our seeing how the uncertainty changes, given more information X.
- We want to select the attributes that give us the lowest conditional entropy, because we want the least amount of uncertainty in our model.
- Picking splitting attributes in decision trees
  - 1) Calculate entropy for the total dataset. This will be the probability of the class * log (probability) and you can get those probabilities from the number of occurrences in the dataset.

    - 2) Pick an attribute, and calculate the conditional entropy for each of the values that the attribute can take on. To calculate that, for each value the attribute can take on, you take the frequency probability of getting that value and multiply it by the entropy of the class labels in that bin. Sum all of them up and the result is the conditional entropy for the attribute.
    - 3) Compute the total entropy - conditional entropy to find the informational gain for that attribute
- When creating a tree, we look at the dataset, and recursively pick the attributes to split on. You stop when all the samples for a given node belong to same class, there are no more attributes (majority vote to classify the leaf), or there are no samples left (majority vote in the parent node).
- For continuous valued attributes, you have to determine the best splitting point for the attribute.
    - Sort all the values in increasing order, the midpoint of each pair of values is a possible split point, and the point with the minimum expected information requirement for A is selected as the split-point for A.
- Gain ratio is used to combat the situation where if an attribute takes on a lot of potential values, you will have a situation where entropy is biased towards those attributes with a high number of values.
    - This is an example of another metric to pick the splitting attributes.
- Gini index is another metric
- Choosing the metric to use is a sort of hyperparameter that we get to choose.
- Let’s look at how to change decision trees from classification outputs to continuous outputs. They are called regression trees.
    - Pick attribute for root node that gives you the lowest weighted average variance.
    - Take the average of the partition for the leaf node.
- You can think of regression trees and the branching as partitioning the input space into the different bins we’re separating into.
- Tree vs Linear Models



- Tree pruning can help the tree representation get more compact and can help prevent overfitting.
- Idea behind random forests is related to bagging. The steps are to create N different versions of the original dataset, train N different classifiers, and then combine the classifiers into one top-level model (through majority vote probably).
- This type of ensemble approach is helpful because if every classifier produces some prediction, then the error will be reduced if we use the average of multiple classifiers.

$$var\left(\frac{\sum_i f_i(x)}{t}\right) = var(f_i(x))/t$$

- Variance gets reduced but bias stays the same.
- With random forests, we sample a subset of different examples and a subset of different attributes/features. Then construct trees for each collection of data examples using the selected variables. Then, aggregate the prediction results.
  - Majority vote for classification and average for regression outputs.

<u>10/29 - Week 5</u>

- There is a difference between lazy learning and eager learning.
  - Lazy: Storing data points until a test tuple is given.
    - Less time in training but more time in predicting.
  - Eager: Given a set of training examples, and then constructing a model before receiving the new test data.

- KNN basically works by taking the test point, computing the distance between all the training examples and the test point. Then, pick the K closest points and the returned prediction will be the average of those K points' labels.

11/5 - Week 6

- For density reachable, every point except for the last one will be a core point.
    - Same with density connected except the first one does not have to be a core point either.
- If two points are density reachable, then they are also density connected.
- A cluster is the set of points where all of them are densely connected with each other.
- Noise is the object no contained in any cluster.
- Border points are those are reachable from other core points but they themselves are not a core point.
- K-Means is not great at handling clusters of different sizes and different densities.
- Hard clustering is where each data point is assigned to exactly one cluster. Soft clustering if where each point is assigned with probability to different clusters.
- The probability a new point x of being part of class j is dependent on the prior probability of class j * probability density function of class j evaluated at point x.

11/7 - Week 6

- In the EM algorithm, you can think of it as split into two stages.
    - E-step: Assign all the data points to particular clusters based on the product of the f(x) where f represents the probability density function and w, which represents the prior probability of the clusters. Once the assignment is done, then I think you can modify w based on that results of the assignment.
    - M-step: Once the assignments are made, then adjust the parameters for the probability density functions for each cluster based on the points that were assigned to it.

11/19 - Week 8

- Absolute and relative support give you measures of frequency for each itemset
    - You can also have itemsets with just one element.
- From all the information, we can derive rules X -> Y where they are both itemsets and you have a certain minimum support and confidence.
    - Support is the probability that a transaction contains union of X and Y.
    - Confidence is the conditional probability that a transaction having X also contains Y.
- A pattern is something that exists frequently in the dataset.
- When talking about support in the context of closed and max patterns, then support is quantified by the count of how many transactions in the database contain the elements in your pattern.

- Finding all the frequent pattern candidates is way too computationally expensive, so there are different algorithms.
- The Apriori property is that every non empty subset of a frequent itemset must also be frequent.
    - Therefore if there is an itemset that is not frequent, then any superset of that set has to be not frequent as well.
- Apriori pruning: Scan DB to get frequent 1-itemsets. Generate length k candidate itemsets from length k-1 frequent itemsets. Then, test the candidates against the DB, and terminate when no other candidates can be generated.
    - Basically, we go from the frequent itemsets of length k-1 to the candidate itemsets of length k and then go to DB and count and get the frequent itemsets of length k.

<u>11/28 - Week 9</u>

- Projected databases are where you apply a projection to every example in the database.

<u>12/3 - Week 10</u>

- For ways to represent a document, you can use:
    - One where each item in the vector represents the ith word in the sequence.
        - The length is the # of words in the sequence.
    - Bag of words approach where each item in the vector represents the count of each of the words. Sparse matrix since most of the counts will be zero.
        - The length is the # of words in the dictionary
- Bernoulli distribution is one that takes two values, each with its own probability, p and 1-p. This can then be generalized with the categorical distribution which allows for k different classes, each with their own probability, and they all have to sum to 1.
- Binomial distribution is basically doing the bernoulli experiment n different times, and seeing the distribution for the total number of times you see positive. Multinomial distribution is the extension of that where you repeat n trials of categorical distributions.

<u>12/5 - Week 10</u>

- A topic consists of a list of words and their probabilities, so basically a word distribution.
- Topic modeling is the process of getting topics automatically from a corpus.
- Theta dk tells us the probability that doc d has topic k. This is a document level parameter. And that value with be the parameter used to generate distribution for topic for each position.
- Zn is topic index and w is word index.