CS 422 Midterm 1 Study Guide

Alex Zorzella

January 2005

1 Decision Trees

Decision trees are trees built from nodes that split based on a collection of data. The decision tree algorithm is supervised because it requires labeled data.
Information gain: $IG = H - \sum_{t \in T} p(t)H(t) \rightarrow -p(\text{left})H(\text{left}) - p(\text{right})H(\text{right})$
Entropy: $H = \sum_{c \in C} -p(c)\log_2(p(c)) \rightarrow -p(0)\log_2(p(0)) - p(1)\log_2(p(1))$

Algorithm:

1. Calculate entropy of the full training set H{}
2. Split training data with each feature and calculate the information gain for each split
3. Choose to split on the feature that gives the best information gain
4. Repeat steps 2-4 on each subset with the best IG = H { above } until the entropy H of all the leaves are all 0 or the algorithm runs out of features

2 K-Nearest Neighbors

KNN is a supervised algorithm that requires no training. It's supervised because it requires labeled data to compare unclassified points to. To classify a new sample, the unclassified input is compared to K of its neighbors in an existing dataset. If there are ties in distance between a point and n of its neighbors, all of those points are considered in the vote. If the vote would still be a tie between any number of classes, then the next nearest point is considered and the process is repeated. If the algorithm is still encountering a tie and all of the data points are considered, then the unclassified input is classified as the first class by default.

3 Clustering

Clustering is an unsupervised algorithm because it doesn't consider the classification of a data point when it's grouping them together. The algorithm takes a value k and returns that many clusters of points. There is an optimal number of clusters k which will best reflect the number of clusters in the data which can be found by comparing the accuracy of each value of k and using the most accurate one.
Clustering algorithm:

1. Randomly initialize k cluster centers $\mu_1 ... \mu_k$ from the data
2. Compute the distance between each point x and each cluster center
3. Assign each point x to its nearest cluster center
4. Recompute the new cluster centers based on the point clusters
5. Repeat steps 2-4 until the centers converge

4 Perceptrons

Basic perceptrons can have labels of -1 or 1.
Error driven algorithm:

1. Initialize the weights and bias to 0
2. Compute activation for a a training sample $(\vec{x}_i, y_i)$ where the activation is $\vec{w} \cdot \vec{x} + b$
3. If $\vec{y}_i a \leq 0$, update the weights and biases using $\vec{w} \leftarrow \vec{w} + y_i \vec{x}_i$ and $b \leftarrow b + y_i$
4. Repeat steps 2 and 3 for all training samples
5. Repeat steps 2-4 until convergence or for a fixed number of epochs

This algorithm slowly shifts the decision boundary of a classifier until it has reached or gotten as close as it can to the optimal boundary. The number of epochs and iterations (and thus the number of total updates) it takes to reach said optimal value differ based on the order in which the points are handled.

5 Gradient Descent

Gradient descent follows a simple philosophy: "ABU — Always be updating". While the error driven algorithm updates the weights and biases conditionally, gradient descent updates the weights and biases on every iteration. The update rules are as follows:

$\vec{w} = \vec{w} - \eta \frac{\delta L}{\delta \vec{w}}$

$b = b - \eta \frac{\delta L}{\delta b}$

5.1 Loss Functions

Before continuing, it's important to discuss loss functions. The ideal loss function is a 0/1 Loss Function, but its discontinuity is its downfall. To account for this, there are functions created to approximate the 0/1 Loss Functions: Surrogate Loss Functions. These include the Hinge Loss Function, Exponential Loss Function, Log Loss Function, and Squared Loss Function.