Assignment 3

Gary Saavedra

Gsaavedra3@gatech.edu

Introduction

The algorithms were implemented in Weka. For K-Means the SimpleKMeans algorithm was used. For Expectation Maximization the EM was used.

Chess Data Set

Why is it Interesting?

The first data set used here is a Chess data set with 3196 instances and 36 attributes. Each instance represents a board configuration in the end stage of a chess game. In this data set the attributes are made up of discrete values. It is a binary classification set with each instance being classified as “win” or nowin”. The data set is fairly balanced with 52% “win” classifications and 48% “nowin” classifications. This data set illuminates some differences in the operating principles of the different algorithms. Analysis of this data can be used to improve a player’s performance during the end stages of a chess match. It will allow a player to determine which moves truly matter when attempting to capture an opponent’s king. For this data set a false postive is characterized as a “nowin” being classified as a “win”. A false negative is a “win’ being classified as a “nowin”.

**Tic-Tac-Toe Data Set**

Why is it Interesting?

The second data set used is a complete compilation of all possibilities of a Tic-Tac-Toe end game board. It is a binary classifier with “positive” or “negative” values. Positive indicates a win for the player, and negative indicates a loss. This data set contains 958 instances and 9 attributes. Each attribute represents a position on the board and has three values: ‘x’, ‘o’, or ‘b’ where b indicates a blank. Each instance is an board configuration at the end of a played game where the ‘x’ player has gone first. This data set differs from the Chess data set in a few different ways. First, it is much smaller at about one third the size of the chess data set. Second, it is unbalanced with about 65% “positive” classifications and 35% “negative” classifications. Analysis of this data can be used for improving performance at tic tac toe. Analysis of the data also shows the positive effect going first has on winning a match, as well as the most important locations to occupy on the board.

Part 1

K-Means Clustering

Canopy works best. Needs higher T1 and T2 values. Plot ideas: error vs random seeds, error vs T1 or T2, need visual for clusters

Increasing beyond k=2 only increases error. Often the extra clusters will end up with no class and only 2 clusters will have a classification. K=1 gives the best error because it just classifies everything as the majority class which in this case is about 65% x’s.

In WEKA in the output there is a table which shows the majority value of each attribute.

Usually only needs a few iterations to get best error rate it can get. According to properties the error is monotonically non-increasing meaning with each iteration error should decrease or stay the same.

**Expectation Maximization**

Does a little worse than k means.

There is some variation in error with respect to number of runs. Might make a good graph.

**Part 2**

**PCA**

Variance covered option tells how much of original interesting stuff in the data we want to keep.

Correlation matrix: Diagonal is standard deviation. Matrix is symmetric. Each row reps feature. Each column reps feature. Entries represent correlation between features. 0 means no correlation. The rank represents the amount of variance that attribute covers. Keep the top attributes amounts to keeping the most variance. The addition of the different features tells us the relative importance of each. Higher proportion means its more important.

Use Ranker numToSelect to choose best attributes

Intuitive understanding of why variance is good for classification. If you have attribute with all the same value it provides no information when trying to classify, its variance is zero. When variance is high there are many values to work with for our classifiers.

**Random Projection**

With a Gaussian and 5 attributes error got down to 43% with 13 seed.

**Part 3**

Top 3 attributes give about 39% error with simple k means.

Graph idea: error vs number of attributes