NBA Player Scouting Report using Machine Learning Methodology

Yuming Qiao
University of California San Diego

Abstract

Machine learning algorithms have been proved to be very powerful in constructing mathematical models to do classification task with enough training iterations. In this project, I used three machine learning algorithms, the Nearest Neighbours, Decision Tree, and Random Forests to explore on my three different datasets. The main goal of my entire project is to use machine learning to study the NBA players. My first two datasets are fetched from UCI Machine Learning Repository, and are used to understand and verify my algorithms. My third dataset is my original dataset, which contains the statistics of NBA players in season 2015-2016. All statistics are based on NBA post season report and the labels are based on my understanding of the game. During the training process, I read through the result and modified my labels because some of them were relatively unfair from the feedback. The outcome of the result reveals the meaning behind the players and the game. In addition, it also gives me a more meaningful understanding of these three machine learning algorithms.

1. Introduction

The Nearest Neighbour algorithm is a simple yet very useful algorithm for classification. In this project, I am using the euclidean distance to choose the neighbours. The values of K depends on individuals datasets, and in my case is 5. Since I am using my own NBA Player dataset, the nearest neighbour algorithm is very helpful to visualize what are my labels. I will explain this detail in the following section. Overall, this algorithm works primarily to verify my dataset make sense. The Decision Tree algorithm utilizes the idea of entropy to minimize the error it classified during the tree process, such that every split it choose is the most efficient cut. My third algorithm, the Random Forest algorithm, is a bagging version of decision tree. It implements the same cut as the Decision Tree, and add a voting function to vote out the label. I will use these three algorithms together to train and test my dataset. My main goal is to use algorithms to fix my dataset and then use the result of my dataset to understand NBA player and the algorithms.

2. Method

2.1 Nearest Neighbour Algorithm

In this project, I make use of the power of nearest neighbour concept. This concept aims to pick the most relevant data as a reference, and thus the label of this nearest neighbour is also the reference for the classification result. Usually, the term “nearest” can be calculated by different criteria. For example, in a 2-D dimension, the distance is the euclidean distance:

$$\text{For X} = \{x^{(1)}, ..., x^{(m)}\}\ , \text{Y} = \{y^{(1)}, ..., y^{(m)}\}\ \ \text{distance} = (\sum_{m}^{i=1}(x^{(i)} - y^{(i)})^2).sqrt(m)$$

In my project, I will use the same euclidean distance. Although my dataset are all in high dimension, the euclidean distance is very easy to implement and still output good results. Then the training process can be viewed as the process of plotting data points and labels in high dimension. In the testing process, we calculate votes of the testing instance by its nearest neighbours, and the label with most votes will be the output label.

2.2 Decision Tree Algorithm

Decision Tree Algorithm is to construct a tree with decision nodes. In the training process, algorithm will continually split the tree until it correctly labels all training sample. For every split, the algorithm uses entropy to calculate the homogeneity of a sample:

Entropy(sample) = $\sum_{i}^{m} - p(i) * log2(p(i))$ (m is the attributes of a sample)

Explanation: If the sample is completely homogeneous the entropy is 0 and if the sample is an equally divided it has entropy of 1.

Process:

Step 1: Calculate entropy of the target.

Step 2: The dataset is then split on the different attributes. The entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split. The resulting entropy is subtracted from the entropy before the split. The result is the Information Gain, or decrease in entropy.

Step 3: Choose attribute with the largest information gain as the decision node, divide the dataset by its branches and repeat the same process on every branch.(Here the algorithm want to choose the split that can best decrease the entropy of the sample)

Step 4a: A branch with entropy of 0 is a leaf node.

Step 4b: A branch with entropy more than 0 needs further splitting.

Step 5: The algorithm is run recursively on the non-leaf branches, until all data is classified.

The decision tree will have a training error of 0.

2.3 Random Forest Algorithm

The Random Forest Algorithm uses the same step as the Decision Tree algorithm. In addition, it will add a voting process when evaluating the labels. In order to add this voting, it will fetch a portion of the training sample to make a few “fake” training samples. Then for each individual training samples, it will perform decision tree algorithm to get the resulting label. It will sum up the votes and pick the most voted one for the output.

The random forest algorithm can reduce the misclassification risk because it performs a “democracy”. In my project, the Random Forest indeed yields a more accurate prediction than Decision Tree.

3. Experiments

NN - Nearest Neighbour Algorithm
DT - Decision Tree Algorithm
RF - Random Forest Algorithm

3.1 Wine Dataset (178 x 13) (Source: See citation page)

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

Algorithm	Train vs Test	Training Accuracy	Testing Accuracy
NN	**0.4 : 0.6**	**0.70**	**0.71**
NN	**0.6 : 0.4**	**0.73**	**0.81**
NN	**0.8 : 0.2**	**0.78**	**0.77**
DT	**0.4 : 0.6**	**1**	**0.92**
DT	**0.6 : 0.4**	**1**	**0.94**
DT	**0.8 : 0.2**	**1**	**0.97**
RF	**0.4 : 0.6**	**1**	**0.97**
RF	**0.6 : 0.4**	**1**	**0.97**
RF	**0.8 : 0.2**	**1**	**0.97**

This is a small dataset that is used as a validation to my algorithms. As we can see, the Random Forest algorithm has the best result. The Nearest neighbour algorithm is not good in this dataset, I propose that it is because all attributes are flat, and using euclidean distance cannot separate them much.

3.2 White Wine Quality Dataset (4098 x 11) (Source: See citation page)

This is a much larger dataset than the previous Wine dataset. There are 11 attributes in this dataset and they are all based on physicochemical tests. The label is the quality of the wine, from 0 - 10.

Algorithm	Train vs Test	Training Accuracy	Testing Accuracy
NN	**0.4 : 0.6**	**0.45**	**0.32**
NN	**0.6 : 0.4**	**0.57**	**0.46**
NN	**0.8 : 0.2**	**0.54**	**0.41**
DT	**0.4 : 0.6**	**1**	**0.78**
DT	**0.6 : 0.4**	**1**	**1**
DT	**0.8 : 0.2**	**1**	**1**
RF	**0.4 : 0.6**	**1**	**0.97**
RF	**0.6 : 0.4**	**1**	**0.98**
RF	**0.8 : 0.2**	**1**	**0.98**

As we can see, the dataset is not friendly with the nearest neighbour algorithm, I think it also because of the same problem that the data is too flat and purely calculating the euclidean distance to determine the label is not wise in this dataset. I think overall because of the specialty of chemicals in wine, using euclidean is not a good practice for wine classification. Similar to the first dataset, Decision Tree and Random Forest have a very good result, and the random forest is sure to have a better result than the decision tree.

3.2 NBA Player Regular Season 2015 - 2016 Dataset (476 x 21) (Source: my own dataset)

Algorithm	Train vs Test	Training Accuracy	Testing Accuracy
NN	**0.4 : 0.6**	**0.79**	**0.77**
NN	**0.6 : 0.4**	**0.8**	**0.77**
NN	**0.8 : 0.2**	**0.82**	**0.8**
DT	**0.4 : 0.6**	**1**	**0.82**
DT	**0.6 : 0.4**	**1**	**0.84**
DT	**0.8 : 0.2**	**1**	**0.87**

RF	**0.4 : 0.6**	**1**	**1**
RF	**0.6 : 0.4**	**1**	**1**
RF	**0.8 : 0.2**	**1**	**1**

I need to clarify that the dataset has been modified by me along with the testing of my algorithms, because my first labeling are based purely on my experience and sometimes is not fair. Although the Nearest Neighbour Algorithm is only has 80% of testing accuracy, but I think this is the most important in this project. The "misclassification" is the most valuable point when we are evaluating a NBA player. Think about this, a player that got 3 votes for Starter, and 2 votes for Substitute from testing, but the actual label is Substitute, what does this mean? This means that this player, from the statistical (efficiency) point, is very controversial between a starter and a substitute. This also means that this player may have the same ability as a starter. The coach, after reviewing this report, can put this player as a starter and try the result. For example, Carmelo Anthony was labeled 2 by me because he is a very good player but was not selected into the NBA First Team (label 1) by NBA people. However, my study result gives him a 1 because he is more close to the people in label 1. Thus, in my study, Anthony deserves a First Team.

4. Conclusion

In my project, I used the Nearest Neighbour algorithm, Decision Tree algorithm, Random Forest algorithm on my three dataset. The Nearest Neighbour algorithm is the easiest and fastest algorithm, and can give very helpful feedback. But using purely the distance is not complex enough in many cases, thus the error rate is the highest. The Random Forests is the most complex and slowest, however, but using voting policy, it can boost the result of Decision Tree. Thus it has the less error rate. The Decision Tree is a decent choice for the dataset. It's speed and difficulty is between the Nearest Neighbour and Random Forest algorithm.

I have done thoroughly experiments in tuning hyperparameters and performing cross validation as you can see in my result table. The uniqueness in my project is my NBA Player dataset. I experiment the process of constructing a dataset, labeling and relabeling based on feedback. **I conclude that sometimes, we need the machine learning algorithm to produce incorrect classification. The inaccurate production can even be more valuable to our study.**

Bonus Point

In this project, the topic I choose to construct my own dataset. This is a very time-spending project, as I have to think of my labels. Moreover, I follow my algorithm to revise my label when I notice something is wrong during the first labeling. I learn a lot during this

construction process, and I will keep on my effort to contribute more result later on with this dataset. I think I deserve some bonus points.

Reference

1. Wine Dataset, http://archive.ics.uci.edu/ml/datasets/Wine, Forina, M. et al, PARVUS - An Extendible Package for Data Exploration, Classification and Correlation. Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno, 16147 Genoa, Italy.
2. White Wine Quality Dataset, http://archive.ics.uci.edu/ml/datasets/Wine+Quality, Paulo Cortez, University of Minho, Guimarães, Portugal, http://www3.dsi.uminho.pt/pcortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis, Viticulture Commission of the Vinho Verde Region(CVRVV), Porto, Portugal
3. Helps for my own dataset, http://www.nbaminer.com/