Chelsea F.C. Game Result Prediction Algorithm

Koray Can Yurtseven
School of Engineering
Middle East Technical University
Ankara, Turkey 06530
Email:koray.can.yurtseven@gmail.coml

Evrim Efe
School of Engineering
Middle East Technical University
Ankara, Turkey 06530
Email: evrimefe12@live.com

Tansel Tunc
Ankara, Turkey 06810
Email: tanselsitkitunc@gmail.com

I. Introduction

In this project a machine learning algorithm is designed which is going to classify the matches of Chelsea F.C against the 3 clubs relegated at the end of each season in the premier league as a win,draw or loss. Chelsea F.C. is one of the biggest clubs of the city of London. The reason Chelsea Football Club has been chosen for this project is that they have been one of the most stable clubs in the last 15 seasons in the chairmanship of Roman Abramovich since the club was bought by Roman Abramovich 16 years ago and the club is a good representative of England. For Chelsea F.C. the past two decades have seen sustained success, with the club winning 21 trophies since 1997. In total, the club has won 27 major trophies; six titles, seven FA Cups, five League Cups and four FA Community Shields, one UEFA Champions League, two UEFA Cup Winners Cups, one UEFA Europa League and one UEFA Super Cup. On the other hand the three clubs that are relegated from premier league at the end of each season usually carry similar features when compared to the major clubs of England and are called yo-yo clubs. Although Chelsea have usually won the matches they have played against the clubs relegated that season, the points lost against these clubs play an important role in the title race. Although soccer game result prediction algoritms exist in the market like betistuta.com these algorithms only take into account the pre-game statistics and these algorithms are not accurate. They only try to help bet players with their predictions. If one of these soccer game prediction algorithms which only take into account pre-game statistics was very accurate there would be betting billionaires. These soccer game prediction algorithms which only take into account pre-game statistics are not very accurate because it is impossible to foresee the result of a game without taking into account during-game statistics. There is even a saying in the soccer world about this phenomenon: ”A game can not be won before it is played” or ”A game can not be won on the paper”. In our classification algorithm both pre-game and during-game statistics are used to come up with a better prediction algorithm. With such a classification algorithm Chelsea F.C. may know what kind of a game strategy they should employ to keep from losing points against the clubs presumed to be relegated at the end of the season. For example if the total number of shots and ball possession of Chelsea are low up until that moment during a game and our algorithm suggests that Chelsea can not win the game when the game statistics up until that moment are plugged in our algorithm Chelsea may know how many more shots they need to take and how much they should improve their ball possession to win the game. In our classification algorithm the features to be used are total number of shots of each team in a match(since as the number of shots of a team increases the teams chances of winning the game increases), shots on goal of each team in a match(since as the number of shots on goal of a team increases the teams chances of winning the game increases), which team is the host(the host team has a higher chance of winning the game with the support of their fans), the number of corners of each team in a match(since as the number of corners a team takes increases the teams chances of winning the game increases), ball possesion percentage of each team in a match(since as the ball possesion percentage of a team increases that team controls the game more and gets a higher chance of winning the game) , which team has scored the first goal(when a team scores the first goal in a game they increase their chance of winning the game), the number of red cards of each team in a match(in the case of a red card a team loses one of its players and decreases its chance of winning the game), betting odds in the form of wins, draws, losses(since these betting odds reflect the collective information on the result of the game), the total points each team has collected in the 10 last games they played against each other, (since this statistic is going to show the recent power balance between two teams), total points collected in the last five league games(this statictic approximates the recent overall power of the team). The dataset obtained between the years 2003-2015 are going to be used as the training data and the dataset obtained between the years 2015-2018 are going to be used as the test data. The datasets are to be obtained from espn.com.

The dataset used in this project is non linearly seperable. Since there are three possible outcomes of a match there are three labels and this is a multiclass classification problem. We are evaluating matches of Chelsea F.C. against the three relegated clubs each season and since Chelsea F.C. is much stronger than these three relegated clubs, Chelsea F.C mostly win their games against these clubs and we have an imbalanced dataset. The definiton of our problem in the literature is Nonlinear Multiclass Classification problem with imbalanced dataset.

We have found 5 mainstream methods about Nonlinear

Multiclass classification problem. These are : ANNs and backpropagation, Bayesian Learning Approach, Knn algorithm, Decision Tree Learning and Support Vector Machines. Support vector machines were originally designed for binary classification. Currently there are two types of approaches for solution of nonlinear multiclass problems using SVM. One is by constructing and combining several binary classifiers while the other is by directly considering all data in one optimization formulation.[1]The two methods based on binary classifcation are oneagainst-all and one-against-one. In one-against all k SVM models are constructed where k is the number of classes. The ith SVM is trained with all of the examples in the ith class with positive labels, and all other examples with negative labels. After solving k dual problems k decision functions are obained and it is concluded that an unseen sample is in the class which has the largest value of the decision function.[2] In the one-against-one method used in [3], classifiers are constructed where each one is trained on data from two classes. There are different methods for doing the future testing after all classifiers are constructed. In [4] the max wins voting strategy is suggested. According to this strategy if the binary classifier constructed for the i, j classes says a test sample is in the ith class, then the vote for the ith class is added by one. Otherwise, the jth is increased by one. Then we predict the test sample is in the class with the largest vote. There exist two all-together methods for solving nonlinear multicslass problems using SVM. One approach was proposed in[5]. In this approach the idea is similar to the one-against-all approach. It constructs two-class rules where the mth decision function separates training vectors of the class m from the other vectors. Hence there are k decision functions, where k is the number of the classes, but all are obtained by solving one problem. In [6], Crammer and Singer also proposed an SVM based approach for nonlinear multiclass problems by solving a single optimization problem. Decision tree algorithms are a very effective technique for classification type learning. The algorithm can achieve good accuracy rates for both binary classification and multiclass classification problems. In multiclass classification, a leaf node can represent either of the K classes in the dataset. To create a tree, one can pick the most valuable attribute using statistical methods. Until reaching a leaf level, one repeats this selection and creates branches and expands the tree. In finding leaf nodes, one must be careful to not overfit their training data. One very widely used method in decision trees is the C4.5 method. In this method, reducing the size of the decision tree to avoid overfitting is the key idea. The main problem is, if one increase the pruning level, the accuracy on the training set will be lower. Nijhawan et. al. [10] claimed that using C4.5 method in WEKA[11] tool, they have achieved better accuracy than using the ID3 method, which is an iterative, simple method. Apart from C4.5, many other methods are proposed in order to improve the accuracy. One proposed method is using C4.5 and one versus all method in multi-class datasets, as it is proposed in Polat and Gnes article[12]. This proposed method was also used by Ramanan et. al[13]. However, due to imbalanced datasets, this method might not be the best decision, because it is cited that it can perform adversely in imbalanced datasets. Therefore combining C4.5 and one versus all method is not suitable for our work. Using other decision tree algorithms and pruning techniques can solve imbalanced dataset problem as well as overfitting. Artificial neural networks have their roots in statistical pattern recognition, and are a generalization of logistic regression. Currently, logistic regression and artificial neural networks are the most widely used models in biomedicine, as measured by the number of publications indexed in Medline: 28,500 for logistic regression, 8500 for neural networks, 1300 for k-nearest neighbors, 1100 for decision trees, and 100 for support vector machines. The quality of the results obtained using an ANN mainly depend on three factors: the quality of the dataset employed, the care with which the adjustable model parameters were chosen, and the evaluation criteria used.[18] Deep neural networks containing multiple non-linear hidden layers offer a great deal of flexibility which allows them to express very complex relationships between the input and the output neurons, however, this also means that they carry a higher risk of overfitting.[19][20] In this case, the model will adapt to the limited training data too well (by memorizing all the cases), and the model will perform poorly on held-out test data.[20] Several methods have been developed for reducing this effect, such as k-fold cross-validation and L2 regularization. Dropout is a technique that prevents overfitting by temporarily suppressing a percentage of the units in a network, thereby preventing complex co-adaptations on the training data.[19][20] The motivation for this comes from a theory of the role of sex in evolution, which states that the ability of a gene to work well with another random set of genes makes it more robust.[19] It has been observed that a dropout of 50 percent of the hidden units and 20 percent of the input units improves classification.[20] Another difficulty in deep neural networks using a Gradient Descent algorithm such as Back-propagation is the Vanishing Gradient Problem. Here, the problem is that, the partial derivative of the Error (Loss/Cost) function with respect to the weights in the early layers is a proportional to the derivatives of all the neurons after it. Thus, if any one of the become saturated (such as when the output value of a neuron with sigmoid activation function becomes too close to 0 or 1), this derivative becomes very small, so it takes a very large number of epochs for some of the hidden layers to get trained properly.[21] A possible remedy is the introduce Time Constants influencing changes of unit activations. [22] Stochastic Gradient Descent can be a remedy to some of the problems introduced by Batch learning. One advantage is that, the noise introduced by each individual data point may help avoid falling into a local minimum. Also, stochastic learning is often much faster and often results in better solutions. On the other hand, the convergence conditions are better understood in Batch learning.[23] Weight initialization has a profound impact on the performance of a neural network. Poorly initialized networks cannot be trained using momentum and well-initialized networks perform markedly

worse when the momentum is absent of poorly tuned.[24]

Imbalanced data typically refers to a problem with classification problems where the classes are not represented equally.In most classifier learning algorithms which assume a relatively balanced distrbution test samples belonging to the small classes are misclassified more often than those belonging to the prevalent classes. In certain applications, the correct classification of samples in the small classes often has a greater value than the contrary case.[7] However in our project correct identification of each class is equally important. Solutions for dealing with imbalanced datasets exist at both data and algorithm levels.[7] SVMs are believed to be less prone to the class imbalance problem than other classification learning algorithms, since boundaries between classes are calculated with respect to only a few support vectors and the class sizes may not affect the class boundary too much. Nevertheless, there are researches which still indicate that SVMs can be ineffective in determining the class boundary when the class distribution is too askew.[7] For SVMs, proposals such as using different penalty constants for different classes, or adjusting the class boundary based on kernel-alignment ideal are reported. [7] Penalized classification imposes an additional cost on the model for making classification mistakes on the minority class during training. These penalties can bias the model to pay more attention to the minority class. For SVMs active learning has also been investigated to be a way of dealing with imbalanced dataset. [8] In SVMs the imbalance ratio of the classes within the margin is much smaller than the class imbalance ratio of the entire dataset. In SVM based active learning the strategy of selecting instances within the margin addresses the imbalanced dataset classification very well. For classification problems, a backpropagation network approximates the class probabilities given the feature vectors of the samples to be classified. Empirical studies on training imbalanced two class data sets observed that the net error for samples in the majority class was reduced rapidly in the first few iterations but the net error for the minority class increased considerably instead.[7] For decision trees, one approach is to adjust the probabilistic estimate at the tree leaf, another approach is to develop new pruning techniques.[7] Solutions at the data-level include many different forms. One of these forms is to resample the original training dataset, either by over-sampling the minority class or undersampling the majority class until the classes are approximately equally represented.[8] One over-sampling method is given as SMOTE in [9]. In SMOTE the minority class is oversampled by creating synthetic examples. In this method the k nearest minority class neighbors of all minority class instances are identified and synthetic minority class examples are created and placed randomly along the line segments joining the k minority class nearest neighbors. In the case of multiclass classification problems the methods tackling the class imbalance problem of binary applications are not directly applicable. For binary-class applications, solutions at data level typically change the class size ratio of the two classes and run the learning algorithm many times in search of the optimal distribution. When multiple classes are present, these solutions are not practical anymore due to the increased search space. Solutions at algorithm level try to adapt the learning algorithms to bias towards the smaller class. When several smaller classes exist, the situation becomes complicated in adapting the learning algorithm. In the presence of data with imbalanced class distributions, another crucial problem especially with the one vs rest approach is that one class versus the other classes will worsen the imbalanced distribution even more for the small classes. To advance the classification of multiclass imbalanced data, a cost-sensitive boosting algorithm AdaC2.M1 is reported in [7].

As the feature dimension of the classification problem increases, so does the systems sensitivity to imbalance. [7] The classification problem in our project has 18 features therefore it is quite possible that the classification algorithm is going to be sensitive to the imbalance in our dataset.

May 06, 2018

II. Methodology

The small number of samples in high dimensional data settings cause the classification model to overfit to the training data, thereby having poor generalization ability for the model.Two of the more common approaches to addressing these challenges of high dimensional spaces are reducing the dimensionality of the dataset or applying methods that are independent of data dimensionality. However we don't want to reduce the dimensionality of our formulation since soccer is a very complex phenomenon requiring may be thousands of variables to explain its dynamics. One of the methods that perform well in high dimensional problems is SVMs.[14] Even though decision tree learning performs well in imbalanced datasets since our problem has a high number of dimensions and because of ease of application we choose SVMs for solving our classification problem. We design our methodology such that it deals with the inherent imbalanced dataset with the highest testing accuracy possible. Before constructing the SVMs we preprocess our data. Preprocessing is done by converting categorical attributes into discrete attributes and by performing scaling. Scaling is done to avoid attributes in greater numeric ranges from dominating those in smaller numeric ranges. Another reason for scaling is to avoid numerical difficulties during the calculation in Kernel SVMs. Because kernel values usually depend on the inner products of feature vectors large attribute values might cause numerical problems.[16] Feature scaling can be done in two ways: 'hard' normalization is mapping the min and max values of a given dimension to 0 and 1 or to -1 and 1. In 'soft' normalization the mean of the values is subtracted and each value is divided by the standard deviation. According to [17] soft normalization yields better results with SVMs therefore we use soft normalization. We deal with the imbalanced dataset both at the data level and algorithm level. At the data level we apply oversampling with SMOTE instead of undersampling since we have few data. At the algorithm level we assign different class weights for the penalty parameter of the SVMs. There exist

two methods to handle nonlinear multiclass problems with SVMs. These are binary classification based oneagainst-all and one-against-one methods. Even though as stated in [15] and [7] the class imbalance becomes even more dramatic in one vs rest approach we use both one vs rest and one vs one methods. We use four different kernel functions(linear, rbf, sigmoid, polynomial) which map the data through a particular nonlinear transformation into a higher dimensional feature space. To make a choice between one vs one method and one vs rest method and among four different kernel functions we are going to do hyperparameter optimization. We employ 3 fold cross-validation to finalize our hyperparameter optimization.

REFERENCES

[1] A Comparison of Methods for Multiclass Support Vector Machines
[2] Comparison of classifier methods: a case study in handwritten digit recognition
[3] Pairwise classification and support vector machines
[4] Another Approach to Polychotomous Classification
[5] V. Vapnik, Statistical Learning Theory
[6] On the learnability and design of output codes for multiclass problems
[7] Classification of Imbalanced Data: A review
[8] Learning on the Border: Active Learning in Imbalanced Data Classification
[9] Smote: Synthetic minority over-sampling technique
[10] The Analytical Comparison of ID3 and C4.5 using WEKA
[11] WEKA: A Machine Learning Workbench
[12] A novel hybrid intelligent method based on C4.5 decision tree classifier and one-against-all approach for multi-class classification problems
[13] Unbalanced Decision Trees for Multi-class Classification
[14] High Dimensional Data Classification
[15] Multi-Class Support Vector Machine
[16] A Practical Guide to Support Vector Classification
[17] https://www.quora.com/SVM-performance-depends-on-scaling-and-normalization-Is-this-considered-a-drawback
[18] https://www.sciencedirect.com/science/article/pii/S1532046403000340
[19] http://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf
[20] https://arxiv.org/pdf/1207.0580.pdf
[21] http://neuralnetworksanddeeplearning.com/chap5.html
[22] http://www.bioinf.jku.at/publications/older/ch7.pdf
[23] http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf
[24] http://www.cs.toronto.edu/ hinton/absps/momentum.pdf