Predicting the Outcome of Scrabble Games

Thomas Brus
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
t.a.brus@student.utwente.nl

ABSTRACT

Scrabble is a board game which has increased in popularity over the last couple of years due to digital variants such as *Wordfeud* and *Words with Friends*. In this proposal we will look at how Scrabble game outcomes can be predicted during the course of the game. In particular, we will look how the likelihood of each outcome can be estimated. We believe this could be an interesting addition to online and mobile Scrabble applications. Several approaches are proposed and eventually compared using a scoring rule called the Brier score.

Keywords

Scrabble, probability distribution, k-nearest neighbors, machine learning, Brier score, game prediction

1. INTRODUCTION

Scrabble is a game in which a combination of skill and luck is required to be successful. Besides having a wide vocabulary, players also need to possess a certain insight or play vision which helps them to recognize what words can be placed on the board. Finally, players need to make tactical decisions and think ahead.

In this research we will examine how to predict the outcome of Scrabble games. More precisely, not the outcome, but the probability of each outcome is estimated. Since the research is focused on two-player Scrabble games, estimating the probability of either one of the players winning will be sufficient[1]. We believe that it is valuable to predict not just the outcome but also its likelihood since this supplies an end user with more information as to which player might win.

There are several approaches that come to mind on how to estimate these probabilities. A naive approach would be to always predict that the player that is ahead is the player that wins. Estimating the probability that a player wins could then be done by picking either zero or a hundred percent based on the predicted outcome.

Another approach would be to look at past Scrabble games and find a situation that closely resembles the situation for which we wish to make a prediction. By not picking one past situation, but multiple, the distribution of past outcomes could be used to construct a probability distribution.

Using machine learning techniques, more advanced models could be built based upon past Scrabble games. One such a technique is a multilayer perceptron. This is an implementation of an artificial neural network, which describes a system of interconnected artificial neurons that transform a number of inputs into one or more outputs. It has a wide variety of practical applications, such as handwriting and speech recognition, but one of the drawbacks is that the parameters of a trained neural network are hard to interpret. Since any output produces a value between zero and one, the neural network could be built such that there is one output that represents the probability that a player wins.

Finally, instead of predicting the outcome, the difference in final scores could be predicted. A prediction interval could be constructed around it, indicating the probability that the score difference is within a certain range. This approach, and the other approaches mentioned in this section are explained in more detail in section 2.6.

1.1 Problem Statement

We will first review what the *probability of winning* means from a mathematical point of view and then show that it is infeasible to precisely calculate it.

The experiment we are interested in is the outcome of an individual Scrabble game. In this experiment the strategies of both players are assumed to be fixed. However, the tiles awarded each turn are drawn randomly. The possible outcomes (the *sample space*) are all tuples of final scores: $\{(x, y) : x, y \subseteq 0, 1, \ldots\}^2$. The probability that the first player wins is then defined as $P(X > Y)$.

In order to calculate $P(X > Y)$, all possible ways in which the Scrabble game could end have to be considered. Their respective likelihood, based on the randomly drawn tiles and the decision making progress of the players, has to be taken into account as well. Since each move results in a larger number of combination of tiles that can be drawn, it is impossible to consider all the different ways the game can end. The fact that there over 3 million distinct racks a Scrabble player can start with [4], illustrates this pretty clearly. Furthermore, it is impossible to know what decisions the players would make in any given situation.

1.2 Goal

[1]The rules of Scrabble are such that a game never ends in a tie.

[2]For argument's sake, it is assumed that players cannot end with a negative score even though technically speaking this is possible.

22nd Twente Student Conference on IT January 23rd, 2015, Enschede, The Netherlands.

Our goal is develop a method that can reliably[3] predict, during any point in the game, which player is going to win. More precisely, the method should be able to correctly predict the outcome more than half of the time. Additionally, the method should be able to express its belief in the predicted outcome. This shall be achieved by producing a probability distribution over the two possible outcomes.

The strength of the belief expressed by the method should be in line with what is reasonably expected. For example, during the initial phase of the game the method should be less convinced of its predicted outcome than during the very end. The same goes for situations in which there is a small difference between the current scores of both players versus when this difference is much larger.

Finally we wish to find a metric that indicates the quality of the produced probability distributions, even though the real probabilities are unknown.

2. APPROACH

In this section we will discuss the steps that were taken to calculate probability distributions and how their quality was measured. The code that was written at each step is available online[4].

2.1 Collecting Scrabble data

As indicated, predicting outcomes could be done based on past Scrabble games. This means a set of completed Scrabble games is required and ideally this is a set of games between human Scrabble players with different levels of experience. We have constructed such a set by fetching games from the *Internet Scrabble Club* [1] server. The Internet Scrabble Club (ISC) is a community where players from all over the world can play against each other using a program called *WordBiz*, their desktop client. One of the features of the desktop client is to fetch the games of any registered player. This process was automated by writing a program that directly interacts with the Internet Scrabble Club server.

2.2 Summary of the collected data

The Scrabbla data consists of 60,138 Scrabble turns, derived from 1731 games played between 1609 distinct players. A number of these games were abandoned prematurely. These games have been discarded.

An average Scrabble game consists of 36 turns. At each turn, a player can take one of three actions: placing a word, swapping tiles or passing. In most turns (92%) the first action was taken, followed by swapping tiles (4.5%) and skipping the turn (3.5%).

It seems that the player that starts the game has a slight advantage. This player wins most of the time (54%) and on average scores 6 more points. The average number of points scored per turn is 18. The average total number of points scored per player is 332.

One would say that it is often the case that the player that is currently ahead is also the player that wins. It turns out that this is true. In a majority of the cases (89%) the player that is ahead wins. If only the first twelve turns are considered, then the effect is less noticable (true in 65% of the cases). To put both in perspective, it would have been surprising if less than half of the games were won by the player that is ahead, since only two-player games were investigated.

2.3 Feature extraction

A number of metrics presented in the previous section, such as the average number of points per turn, could not be directly fetched from the collected Scrabble games. These metrics have been calculated and in the context of machine learning this is called feature extraction. Usually, those features are extracted that are expected to have a lot of predictive value.

The following features have been extracted from our data set, for both players: current score, Internet Scrabble Club rating[5], number of turns, average score per turn, number of bingos [6] scored, average number of bingos per turn, the number of blank tiles hold currently and the total value of all tiles on each player's rack.

Additionally, total number of tiles left, the total number of turns and the number of turns divided by the number of tiles left (we will call this the *progress*) have been calculated. Where possible, features have been combined to form new features, such as the average score difference which is the result of subtracting the average score of the first player from the average score of the second player.

Finally, the final score of both players and the outcome (defined as whether or not the first player has won) are extracted. These are required by most machine learning algorithms in order to create a model and, moreover, these will be used to measure the accuracy of the predictions.

2.4 Correlation of features to the outcome

In order to get a sense of which features have a lot of predictive value, it is interesting to investigate how much they are correlated to the outcome. More specifically, the final score difference. The results are summarized in table 1. It is important to note that features with a lower correlation coefficient are not per se useless. For example, a clever algorithm could discover that the combination of a large score difference and a small number of tiles left provide strong evidence for which player has the upper hand.

2.5 Training & test set

The final step taken before making predictions is to divide the collected data into two parts. The first part (the training set) is used to calibrate the machine learning algorithms. The second part (the test set,) is used to measure the performance. In our experiments the training set is twice as large as the test set, since this is common practice.

2.6 Estimating probabilities

In this section, five approaches for estimating probabilities are discussed.

2.6.1 Based on score difference

As explained in the introduction, the simplest method is to either pick a zero or a hundred percent chance that the first player will win based on whether this player is currently ahead. This will not produce the probabilities that we are after, namely probabilities that are nicely divided between zero and one, but we nevertheless wish to investigate this method.

[3]Obviously, no guarantees can be made on the outcome of an individual game, if only because of the role of luck.
[4]https://github.com/thomasbrus/probability-calculations-in-scrabble

[5]http://www.isc.ro/en/help/rating.html
[6]A Scrabble play in which all seven letters are used.

Table 1. Correlation of features to final score difference.

Feature	Coefficient
current_score_difference	0.74
average_score_difference	0.74
second_player_average_score	-0.32
first_player_average_score	0.31
rating_difference	0.23
first_player_current_score	0.21
first_player_number_of_bingos	0.21
second_player_current_score	-0.21
second_player_number_of_bingos	-0.20
first_player_average_number_of_bingos	0.16
second_player_average_number_of_bingos	-0.15
second_player_rating	-0.05
second_player_rack_blanks	-0.04
first_player_rating	0.03
first_player_rack_blanks	0.03
first_player_rack_value	0.02
first_player_number_of_turns	0.01
second_player_number_of_turns	0.01
number_of_tiles_left	-0.01
progress	-0.01
second_player_rack_value	-0.01

2.6.2 *Nearest neighbors*

The nearest neighbor algorithm is arguably the simplest machine learning techniques. The idea is to pick the sample from the training set that is most similar to the sample for which we wish to make a prediction. This sample is then called the *nearest neighbor*. The similarity is calculated by measuring the difference between the two feature sets. All features are first normalized such that they are all within the same range.

The estimated probability is based upon the outcome of the nearest neighbor. Again, either zero or a hundred percent is chosen since there is only one sample available.

2.6.3 *K-nearest neighbors*

This methods improves upon the previous method by not picking a single, but instead k nearest neigbors. Despite its simplicity it can be quite accurate. K-nearest neighbors is usually used for classification tasks, by picking the most prevalent outcome among the selected neighbors. Since our goal is to create a probability distribution, we will do so based upon the distribution of outcomes amongst the neighbors. As an example: if eight out of ten neighbors predict the first player wins, then our algorithm will predict that there is an 80% chance the first player wins in the current game.

2.6.4 *Artificial neural network*

The neural network implementation that we will use is a multilayer perceptron [7]. Artificial neural networks are known to be capable of recognizing non-linear relations. Furthermore, their outputs are always in the range zero to one. We will set up the neural network such that it has one output. In order to make an estimate the value of this output will be used as is. Whether this is justified will show from the results.

An alternative approach would be to predict final scores and construct two prediction intervals. Constructing prediction intervals based on neural network predictions is discussed in length by Khosravi et al. [5].

2.6.5 *Multiple linear regression*

Multiple linear regression is a form of linear regression where multiple explanatory variables are used, as shown in equation 1. Here y is the predicted value and $f_1 \dots f_n$ are the features on which the prediction is based. The weights $\beta_1 \dots \beta_n$ are chosen such that the algorithm is as accurate as possible, using for example *ordinary least squares* (OLS).

$$y = \beta_0 + \beta_1 * f_1 + \beta_2 * f_2 + \ldots + \beta_n * f_n \qquad (1)$$

Using multiple linear regression, the difference in final scores can be predicted. A prediction interval can be constructed around this value (as discussed by Kononenko et al [6] as well as Briesemeister [3]). Prediction intervals define the probability that a future observation lies within a certain range.

We will assume that the predicted final score has a normal distribution. The mean u of this normal distribution equals the predicted final score, but the standard deviation σ is unknown. It is, however, possible to estimate the standard deviation based on the training set and the predicted final score. This was done using the R programming language.

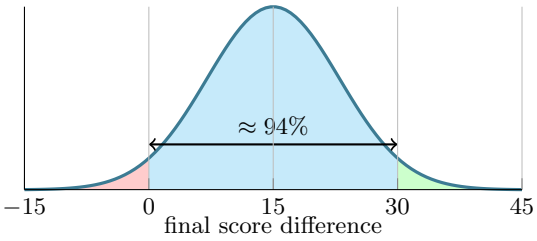

Figure 1. Prediction interval 1 (predicted score difference = 15, estimated standard deviation = 8).

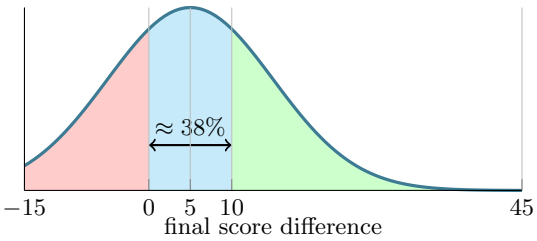

Figure 2. Prediction interval 2 (predicted score difference = 5, estimated standard deviation = 10).

The combination of a mean and an estimated standard deviation allow us to construct a prediction interval, as shown in figure 1 and 2.

In figure 1, a normal distribution with a mean of 15 points and an estimated standard deviation of 8 points is illustrated. The area under the graph that ranges from zero to thirty occupies 94%. This means that it is estimated that the probability of the final score difference being within the interval $(0, 30)$ is 94%. Adding to this the area on the right (which occupies 3%), the estimated probability that the first player wins in this example is 97%.

In figure 2, the effect of the size of the final score difference and the estimated standard deviation is visible. In this

example the estimated probability is only 69% $(0.38+(1-0.38)/2)$.

When the predicted final score difference is negative, the same method is applied, but the probability is inverted.

2.7 Performance analysis

The performance of each approach is measured using two metrics. These are the accuracy and the Brier score [2]. The accuracy is defined as the percentage of correctly predicted outcomes. The Brier score is a metric that was invented to measure the quality of weather forecasts. It is applied to a range of probabilities and a range of corresponding (binary) outcomes. The Brier score is defined as follows:

$$BS = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2$$

Effectively, the difference between the estimated probability (f_t) and the actual outcome (o_t) is squared, then summed over all predictions and finally divided by the number of predictions. The Brier-score acts as a so-called proper scoring function in that it ranks algorithms by how well their estimated probabilities match the true probability. Better estimators have a lower Brier score. This property serves as our motivation for comparing the approaches using the Brier score.

3. RESULTS

3.1 Analysis of predictions

Before presenting the accuracy and Brier scores of the different approaches, we first wish to visualize the estimated probabilities, *regardless of whether they are accurate*. Since the first two approaches estimate either zero or a hundred percent we will ignore them for now.

The other three approaches, however, are much more interesting. In figure 3 till 5, the estimated probabilities versus the number of turns are shown. On the y axis the largest of the two probabilities ($p1$ and $p2$) of both players is showing. It shows that in the initial phase of the game, k-nearest neighbors often predicts probabilities in the 50%-70% area, whereas later on, each probability is chosen about as often. In figure 4 (artificial neural network), quite the opposite is shown since in the beginning all probabilities are chosen about equally often and later on the algorithm tends to pick more extreme probabilities (that are farther away from 50%). In figure 5, it shows that probabilities estimated by multiple linear regression are affected the least by the number of turns taken.

In figure 6 till 8 the estimated probabilities versus the current score is shown. It is interesting to see that the last method (multiple linear regression) shows the strongest relation between the current score difference and its predictions.

3.2 Accuracy & Brier scores

We will now present the main results of our research. As explained in section 2.7 both the accuracy and the Brier score are calculated. In figure 9 and figure 10 these two are shown for all of our methods. The following abbreviations have been used: *Naive* (approach based on which player is ahead), *1NN* (nearest neighbors), *KNN* (k-nearest neighbors), *ANN* (artificial neural network) and *MLR* (multiple linear regression). The horizontal baseline in the first diagram indicates the accuracy achieved by randomly predicting which player wins. The baseline in the second diagram

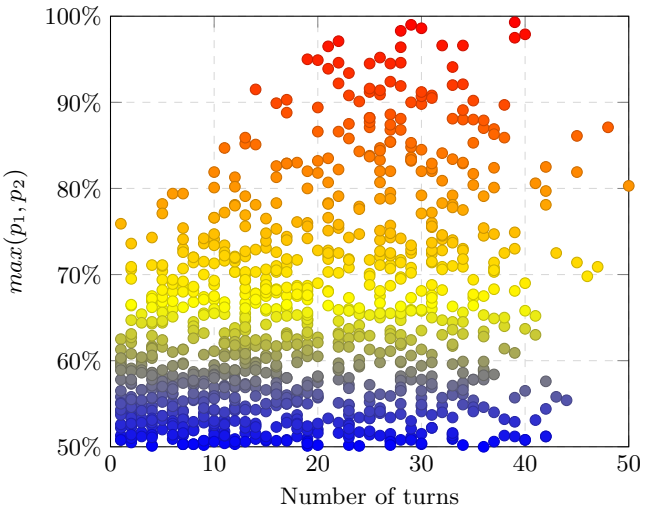

Figure 3. K-nearest neighbors: estimated probabilities vs. number of turns.

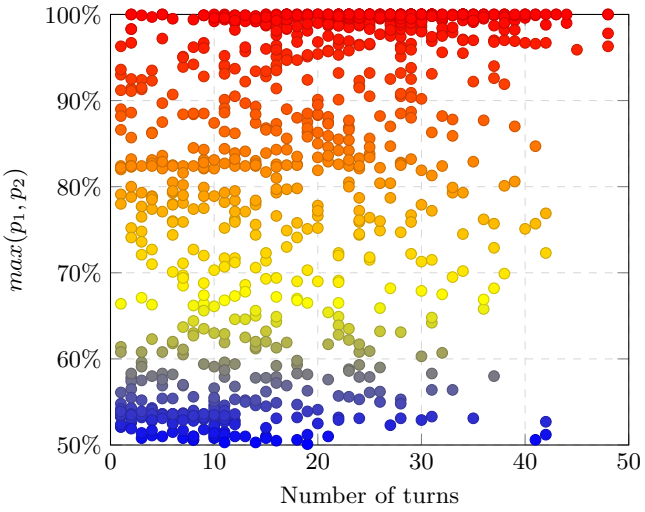

Figure 4. Artificial neural network: estimated probabilities vs. number of turns.

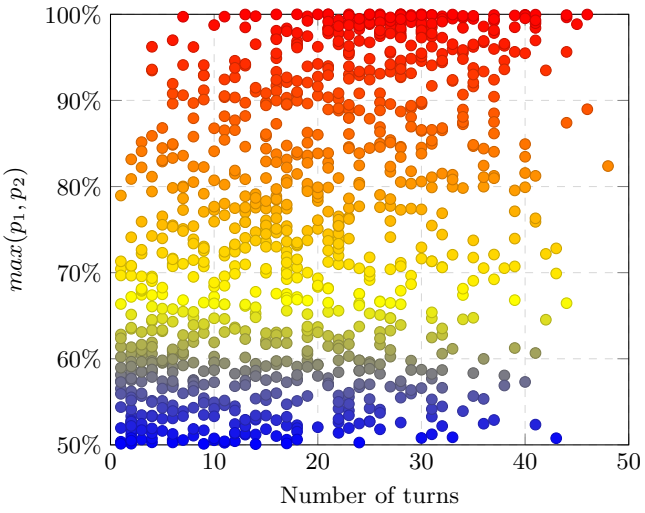

Figure 5. Multiple linear regression: estimated probabilities vs. number of turns.

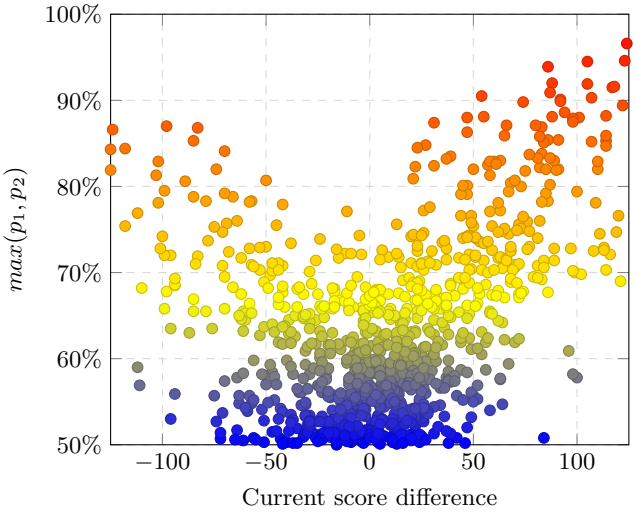

Figure 6. K-nearest neighbors: estimated probabilities vs. current score difference.

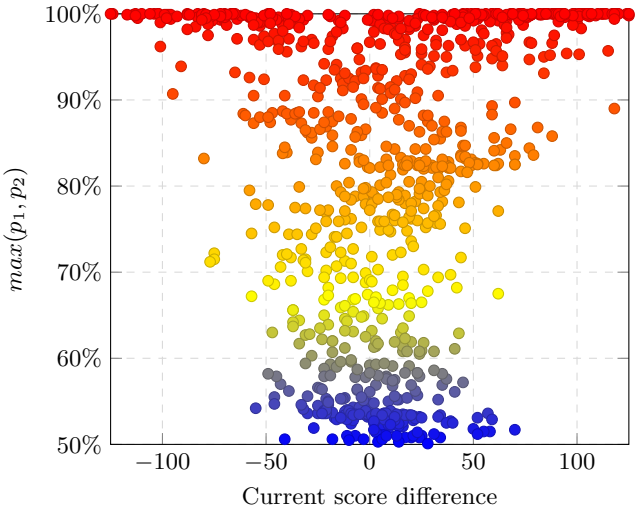

Figure 7. Artificial neural network: estimated probabilities vs. current score difference.

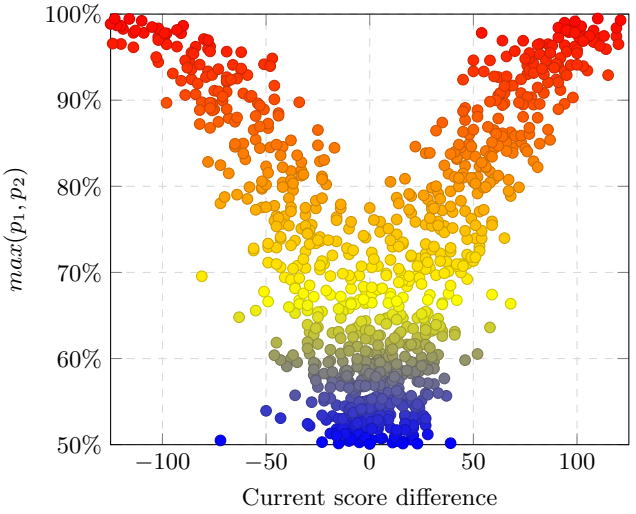

Figure 8. Multiple linear regression: estimated probabilities vs. current score difference.

indicates the Brier score achieved by always estimating a probability of 50%.

Next, the same metrics have been calculated for all the results where the player that was ahead did not win (figure 11 and figure 12). This gives insight in how much the methods rely on the current score difference. Note that the naive method has an accuracy of zero percent as it completely bases its prediction on which player is ahead.

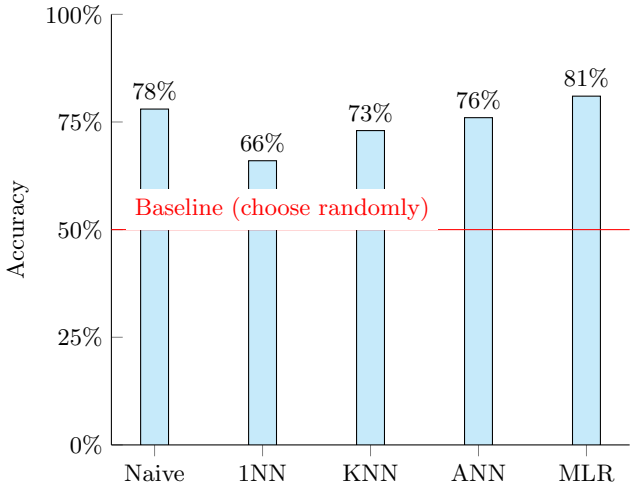

Figure 9. Overall percentage of correctly predicted outcomes.

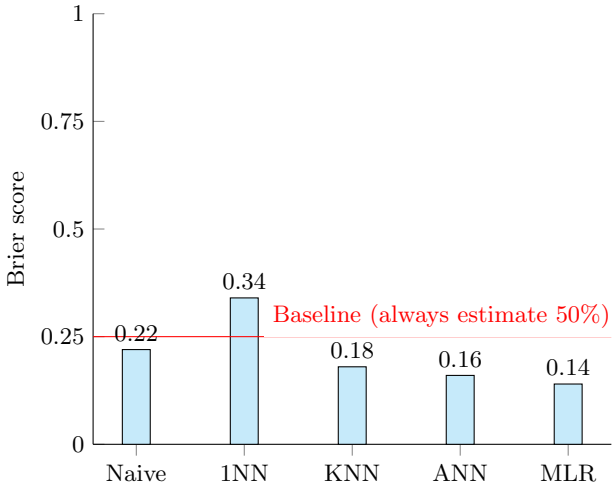

Figure 10. Overall Brier score of predictions *(lower is better)*.

3.3 Interpretation of results

From figure 9 it shows that all algorithms are capable of predicting the outcome of a Scrabble, with a accuracy that is better than randomly guessing.

Estimating the probability of a certain outcome is a different matter. To recap, the algorithm that best estimates the true probabilities shall have the lowest Brier score. From our analysis (section 2.7) it already became clear that the more sophisticated methods (KNN, ANN and MLR), produce probabilities that are more in line with what is to be expected. For example, all three methods showed more outspoken predictions (in the range of 80%-100%) as the number of turns increased. Consider that the

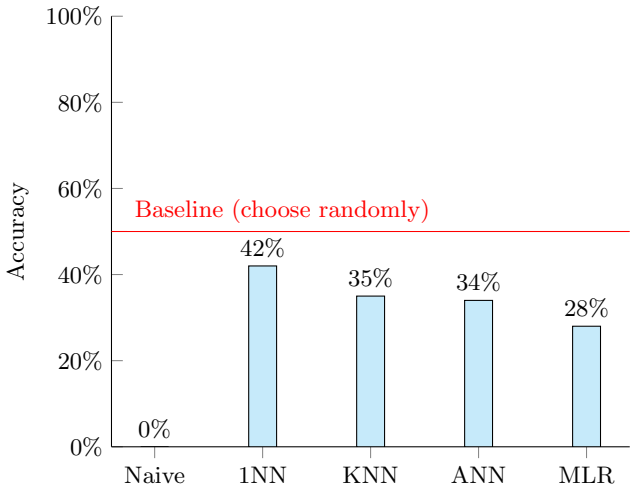

Figure 11. Percentage of correctly predicted outcomes, where the player that is ahead did not win.

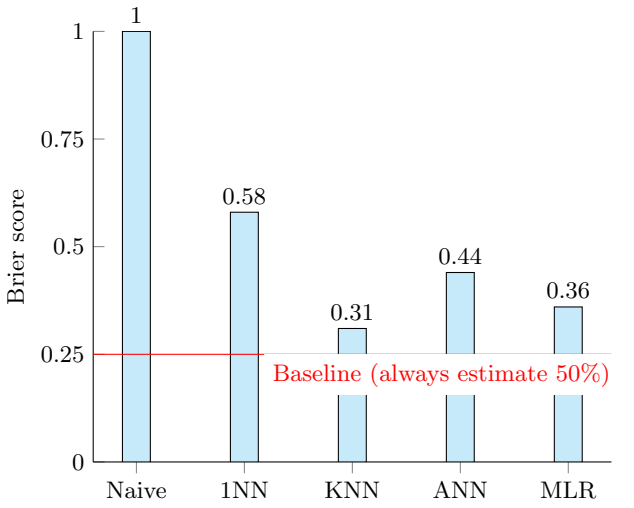

Figure 12. Brier score of predictions, where the player that is ahead did not win *(lower is better)*.

other two methods (naive and 1NN), produced probabilities that are either 0% or 100%. Figure 10 confirms our assumption that the more sophisticated algorithms produce more accurate probability estimates. That is, probability estimates that closer match the true (unknown) probability. Both figures indicate that multiple linear regression performs best.

In figure 11 and 12 a subset of the predictions have been inspected. Namely the predictions that had a surprising outcome: the player that was ahead did not win. It is to be expected that all algorithms have a hard time correctly predicting the outcome, since the current score difference is the feature that has the highest correlation to the outcome (see table 1). Nearest neighbors is the algorithm that is least affected by this feature and it shows from the results in figure 11. The other algorithms make up a little bit by taking into accounts features such as the player's ratings and the number of turns made.

Figure 12 shows that despite the more sophisticated algorithms having poor accuracy, they still succeed in making reasonable probability estimates. This is important since an algorithm may have an accuracy of zero percent (by often being just below or above 50%), but still provide a very realistic estimate of the outcome. Also note that a random probability estimate would result in a Brier score of 0.33. Finally, it is surprising to see that the artificial neural network scores 0.44. The behavior of neural networks is often hard to explain.

In conclusion, multiple linear regression performs best overall, at predicting outcomes as well as at producing probability estimates.

4. DISCUSSION

As indicated in the beginning of this paper, it is impossible to precisely calculate the probability that a certain player wins. This limits our ability to measure the quality of our own predictions but also serves as the main motivation.

The metric we chose to use is the Brier score. Even though the Brier score encourages to estimate probabilities that are close to the true probability, it is still not bullet proof. Theoretically an algorithm could achieve a minimal Brier score by correctly predicting 100% when the first player wins, and 0% when the first player loses. However, since the Brier score squares every error that is made, and since an algorithm shall never be able to predict the outcome correctly 100% of the time, we still feel that the Brier score precisely measures the performance of our methods.

5. CONCLUSIONS

In conclusion, we have found that it is very much possible to accurately predict Scrabble game outcomes. Even though there are literally millions of different possible Scrabble games, and even though luck plays a role, it is still possible to pick the actual winner most of the time. This is shown by the fact that an accuracy of 81% was accomplished which is a vast improvement over picking a winner randomly (accuracy of 50%).

Furthermore, this research has delivered several methods to estimate the probability of a certain outcome. Each method has been measured by a well established scoring function, namely the Brier score. The most advanced method, multiple linear regression, turned out to be the most performant method. Using this method, winning probabilities can be calculated that are accurate, plausible to an end-user and easy to interpret.

6. ACKNOWLEDGEMENTS

I would like to thank Arend Rensink for the helpful comments and discussions throughout the research.

7. REFERENCES

[1] Internet scrabble club. http://www.isc.ro.

[2] G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[3] S. Briesemeister, J. Rahnenführer, and O. Kohlbacher. No longer confidential: estimating the confidence of individual regression predictions. *PloS one*, 7(11):e48723, 2012.

[4] R. K. Hankin. Urn sampling without replacement: enumerative combinatorics in r. *Journal of Statistical Software, Code Snippets*, 17(1):1, 2007.

[5] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya. Comprehensive review of neural network-based prediction intervals and new advances. *Neural Networks, IEEE Transactions on*, 22(9):1341–1356, 2011.

[6] I. Kononenko, E. Štrumbelj, Z. Bosnić, D. Pevec, M. Kukar, and M. Robnik-šikonja. Explanation and reliability of individual predictions, 2012.

[7] Wikipedia. Multilayer perceptron — Wikipedia, the free encyclopedia, 2015. [Online; accessed 4 January 2015].