Final Report Christopher Ho

Does MMA Math Work? A Study on Sports Prediction Applied to Mixed Martial Arts

Introduction

Sports have been a part of human civilization since even before the classical period. But just as sports were invariably a part of society, so too was the desire to know about the future. The ancient Greeks had their Olympics, but they also had their seers. Desiring to know the future of sports is just a natural consequence of these two mainstays of human civilization.

Predicting the outcome of sporting events make for an interesting and not altogether trivial AI and machine learning problem. In spite of the popularity of both AI and sports, most applications were of unsupervised learning to discover interesting new structure in sports. Of sports predictors, there has only been a single paper written: MYMAIT, which used neural networks to predict rugby results. This system achieved 67% accuracy on its initial run, and after 10 years of additional data, this accuracy has increased to 85%.

The lack of work done in this area is a shame, as many modern sports have a glut of statistics for each event, making for an innumerable amount of data with which to predict. Unfortunately sporting events are also very stochastic by nature. Someone might break an arm, someone might have a bad day and the list of valid "if's" go on. Modern mixed martial arts (MMA) is a sport that takes this randomness to an extreme.

In mixed martial arts, as in boxing, it is widely said that when a fighter is completely outmatched on paper, he or she has a "puncher's chance". That statement alone describes not only what makes the sport so exciting, but also what makes it so random for the purposes of prediction. At any given moment in the fight, it only takes a single punch, or slip-up to get knocked out or submitted. Anyone can win at any time. It is because of this that it is widely ascribed among fandom that "MMA Math" doesn't work. Just because Fighter A beat Fighter B and Fighter C lost to Fighter B, it does not necessarily mean that Fighter A beats Fighter C. That math, however, is a simple inequality that does not characterize many aspects of the sport. In this project, we will attempt to make "MMA Math" work.

Challenges

Sports are all about the human element. In this statement lies the majority of the challenge for this problem. A number of factors contribute to human performance at any given time, and nearly all of these factors are hidden from us. Some are unquantifiable, such as emotional or mental state, and others are simply uncharacterized by the available data, such as number of broken bones, or time since last concussion. The vast amount of unmodeled dynamics in sports contribute to a great deal of randomness in the problem, and extremely noisy data.

Predicting the winner in mixed martial arts competitions as a problem suffers from these issues more strongly than other sports. Where in traditional sports, a last minute comeback from behind is a rare, memorable event, in mixed martial arts, where it takes only a single mistake to turn the tides of a match, such comebacks are not uncommon. This contributes to more outliers and randomness in the data. Further, where Combine stats or rushing yards would be in the discussion for NFL players, intangibles such as ‘heart’ or ‘chin’ (ability to take a punch) are often central in discussions about mixed martial artists. While these attributes of a fighter are difficult to characterize by data, they are undoubtedly important factors in determining the winner. This too contributes to the randomness of the problem. Finally, it is important to note that where in other sports an athlete might compete once a week during a season, you would be hard pressed to find a mixed martial artist who competes more than 3 times a year, overall giving less data per fighter.

Approach

To produce quick results, a simple baseline naïve bayes classifier, P(A|B), based on win-loss records was implemented. Here, A was the state of a fighter winning, given B, the record of the fighter, and the record of his opponent. The prior, P(A), then is simply the win rate, while P(B) is the probability of their records, given as the local probability of the win/loss at that point in either fighter’s career (e.g. P(B_i) = P(fighter A wins AND opponent loses) or vice versa, given that both events have to occur). P(B|A) was a recalculation of the probability of a fighter’s record given an extra win on his record.

It is worthwhile to make a digression about the data structure required to make many aspects of this project work. In brief, a lattice structure was created in which there were N+1 instances of a given fighter, where N is the number of fights the fighter has had on record. Each instance of a fighter would correspond to the record and career statistics of the fighter after a given fight, and vice versa. Each fight object contains pointers to the fighter objects containing the statistics of the fighter before the fight, and pointers to fighter objects for the statistics of each fighter after the fight. Similarly, each fighter object has a pointer to his or her last and next fight. This data structure allows us to move backwards in time to characterize a fighter by who he or she has fought and how well he or she has done against given opponents.

While the data was not linearly separable, making the perceptron algorithm a relatively poor choice, it was an excellent starting point to test out a richer set of features. The basic set of features that laid the groundwork for all later variants were statistics culled from Fightmetric.com: height, weight, reach, age, wins, losses, striking accuracy, striking rate, striking defense (% of opponents strikes that miss), and so on. The training and development error was found by averaging the error over different partitions of the data, and taking a subset of each partition to provide error vs. examples. This provides us with a much more reasonable baseline than the naïve Bayes classifier which simply showed that the problem was tractable.

A number of variations of this feature set were attempted, such as averaged relative stats, which took $a_j = \frac{1}{N}\sum_{i=1}^{N}\frac{A_{ij}+0.1}{B_{ij}+0.1}$, as its features, where A_{ij} was the j^{th} statistic of fighter A i-1 fights ago, and B_{ij} was the j^{th} statistic of A’s past (or current) opponent i-1 fights ago. The intention behind this experiment was to attempt to characterize a fighter by his or her past performance relative to his or her opponents. Other features were also used, such as including the specific stats of each fighter’s last fight (the fighters’ previous fight-day performance).

Continuing on with this theme of modifying the feature space, we also attempted to use kernels to better fit the data, using kernelized perceptron, and the one-class SVM (using LIBSVM) to classify the data. While only a simple 2nd order polynomial kernel was used for the perceptron, a number of different built-in kernels were used for the SVM, ranging from linear and polynomial, to radial basis and sigmoid of different degrees.

Because Mixed Martial Arts is a sport with two agents, we also attempted to pose the match as an adversarial game with random elements. This was accomplished by following each fighter's action (strike, takedown, submission) with a dice roll which would bring the game to a new state based on each fighters' prior statistics. Greater rewards were given for definitive wins such as by knockout or submission. Lesser rewards were given for a decision win, which was determined as who had the greater weighted sum of successful strikes and takedowns at a given depth in the game tree. The victory metric was the expectiminimax value of the game, where A (the max agent) would be predicted if the value of the game was positive, and B otherwise. This game value was used not only as a standalone metric, but also as a feature in the classifier. It is important to note that each ply of the game consisted of four layers, A's max node, A's dice roll, B's min node, and B's dice roll, where each step had a branching factor of about 3. This meant that even with alpha-beta pruning, it was only reasonable to run up to a depth of 3 plies, roughly corresponding to the three rounds in a typical fight. The significant depth limitations of the adversarial game motivated the next experiment.

The initial incarnation of the game only allowed a single action per turn. It was clear that it would be more informative if each ply of the game represented a unit of time, over which each fighter would attempt so many of instances of an action based on their striking/takedown/submission attempt rate statistic. While more representative of what might happen in a real fight, this design limited a fighter to only a single action over a 5 minute round, where in reality a fighter could do any combination of the three actions.

Taking this concept to its logical conclusion, we first examined the accuracy of predicting the winner of each fight using a posteriori fight data, that is, data from the fight itself known after the fact. Because we achieved good results from this (83% accuracy over all types of finishes with perceptron), we then used simple linear regression to predict each of these fight-day values using our basic features, and then trained the perceptron algorithm using these estimates as features.

Results

The naïve bayes classifier that was first implemented did not work well. For a small sample of tests, it predicted the winner correctly approximately half of the time. However, when fed fights considered to be among 'The Biggest Mismatches in UFC History' (via an online article), it predicted the correct result at a much higher rate, with a much larger margin than before, showing some validity to the classifier. In all, it is clear from this simple example that a fighter cannot be characterized probabilistically by their win/loss record alone, thus motivating our later experiments.

The perceptron with basic features achieved fair accuracy, ranging from 58% to 68% accuracy on the development sets, averaging to approximately 63% after training on the full training set. While these results are not especially informative alone, we can use this as a fair baseline, and compare it to the literature baseline of 67%. It is important to note that the data is very clearly not linearly separable, nor does it have large margins. As such, the perceptron algorithm is not especially well suited for this task. In spite of this, it does an admirable job. This motivated the use of LIBLINEAR which can handle both smaller margins and inseparable data. Furthermore, it is also worth noting that when testing accuracy was plotted against training set size, as seen in Figure 1, the data varied wildly and a weak linear trend, indicating that the error was dominated by bias, or poor model selection. This result motivated the experiments with different feature sets.

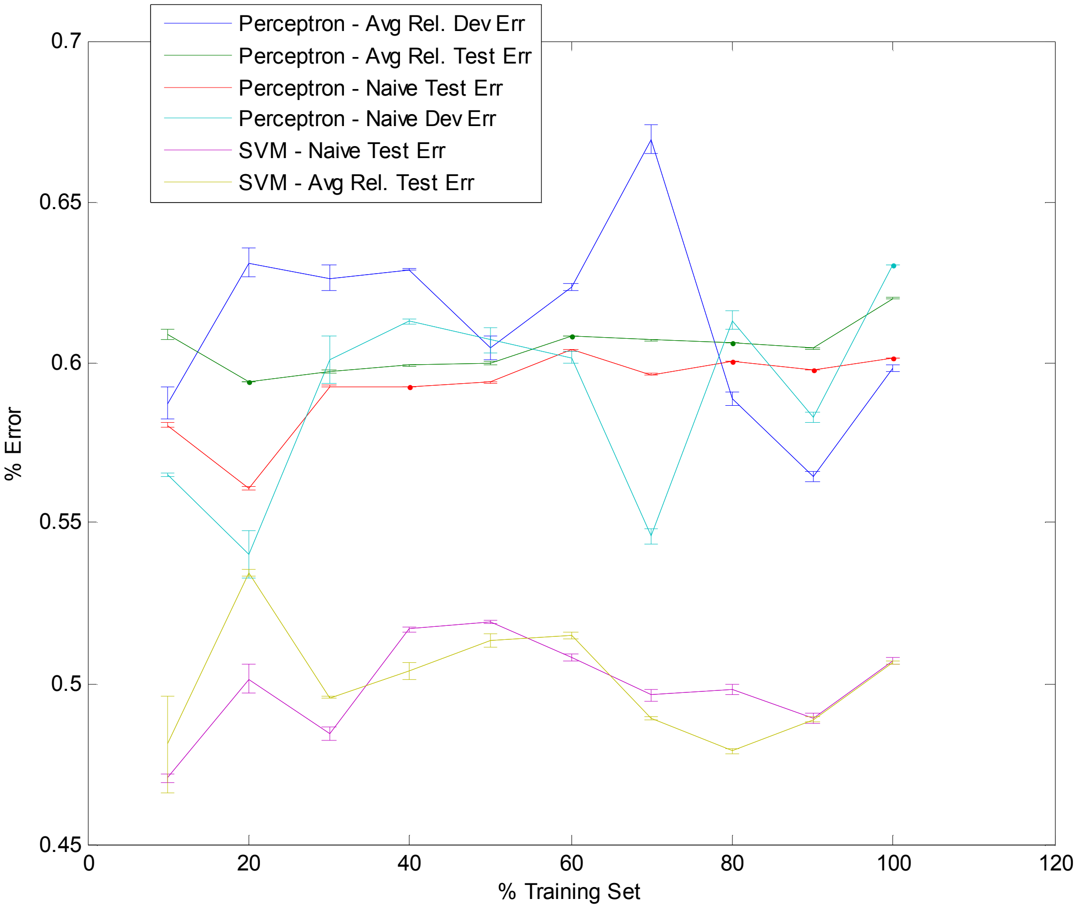

Figure 1: Error of different classifiers

Experiments with different feature sets proved to be a futile gesture. Using averaged relative features achieved similar, but slightly lesser accuracy than the basic features, attaining approximately 60% accuracy. While we were operating with slightly more informative features, it is likely that dividing noisy data by more noisy data amplifies the influence of noise significantly. Further, it is worthwhile to note that where each feature was divided by the same feature on the opponents side (e.g. A striking accuracy / B striking accuracy), it may have been more informative if these quotients were mixed in a more meaningful way, such as pairing offensive statistics with defensive statistics (e.g. A striking accuracy / B striking defense). Using each fighter's performance from his or her last fight did not appreciably affect the testing accuracy.

Both SVM and the kernelized perceptron had very poor results, averaging around 52% accuracy. Oddly enough, hand coding in basic features raised to any power also significantly decreased performance. This is most likely because taking the data to any power other than one would cause large numbers to blow up and small numbers to shrink, vastly increasing the effects of noise on already noisy data. Failing to preprocess the data also likely contributed to these effects.

Modelling the fight as a game simply did not work, achieving an accuracy of approximately 54%. Because of the branching factor of 3^4 = 81 for each ply, it became prohibitively expensive to compute the expectiminimax value of the game tree beyond 3 plies. Three actions per agent is very clearly nowhere near enough to characterize a fight where on average action sequences are of length 100.

Similarly, it was clear that the completely inaccurate game expectiminimax values were not especially informative as a feature. This was similarly reflected by an unchanged training accuracy when including the game value as a feature. Examining the flaws in this approach, however, motivated the separate paradigm that achieved better results.

Using linear regression to estimate fight-day performance, and running the perceptron algorithm with those estimates as features achieved 64% accuracy. While this performance does not seem especially good, it is important to note that our upper limit, using a posteriori data on the perceptron algorithm, was an accuracy of 83%. Factoring in the error in the regression gives us our final accuracy of 64%. Because very simple algorithms used to estimate the fight-day data, and to classify the winner, there is significant room for improvement in this approach, both by improving the regression estimates, and by using a classifier that can better handle smaller margins and nonseparable data.

These results are summarized in Table 1.

Table 1: Classification Accuracy of Different Techniques

Classifier	Training Accuracy	Testing Accuracy
Naïve Bayes	--	50%
Perceptron – Basic Features	60%	63%
Perceptron – Averaged Relative Features	62%	60%
Perceptron—Fight Day Estimates	66%	64%
Kernelized Perceptron	58%	51%
Game	--	54%
Linear Classifier (LIBLINEAR)	--	68%
SVM	55%	53%

Conclusions

The 68% accuracy rate obtained with the linear classifier, was a fair result. While not especially good in the broader realm of classifiers, compared to the one example in literature, MYMAIT, which achieved a 66-67% accuracy using more sophisticated neural network techniques on arguably less noisy data, this classifier does a commendable job.

Future Work

In the future, more sophisticated machine learning techniques will be used. Neural networks appear to be the most promising solution given the relative success of using regression to estimate fight-day stats for a classifier, which was essentially a poor man's neural network. Further, it will be worthwhile to normalize the data to make the effects of kernelization more tractable. Another promising direction to take this project is to use factor analysis or mixture of gaussians to handle the unmodeled effects inherent in the problem. In all, predicting the outcome of mixed martial arts matches is an interesting and difficult problem that we have only scratched the surface of.