Exploring Predictors of Team Success in Ultimate Frisbee: An Analysis of Game Statistics for Stanford Women's Ultimate

Caitlin Go, *cgo2@stanford.edu*

Abstract—One Paragraph on motivation and high-level methods and results

1 Introduction

While Ultimate Frisbee (Ultimate, for short) has long existed as a fun, college pastime, it has grown significantly in recent years as a competitive sport at the youth, college, and adult club levels. As a result, the Ultimate Frisbee community is still struggling to find meaningful ways to gather, create, and analyze game data. See Appendix A for more information on the sport.

Luckily, the Stanford Women's Club Ultimate Head Coach, Robin Davis, has been keeping detailed records of the team's games since 2002. Each game report contains the final score as well as point-by-point information - which team scored a point, turnovers for each team, and specific player statistics for the Stanford Women's Club Ultimate team. As a member of the team who has been granted access to this information, I have the opportunity to more deeply analyze the player and team features that contribute to success in Ultimate Frisbee games.

The first goal of my project is to find data-driven classifications of players and opponents, to understand the variety in the sport and features that drive it. The second goal of my project is to discover if these features can predict team success, and if so, which of these features have the largest impact.

To find the player classifications, the input will be samples of player data per game; these include goals caught, defensive plays made, drops of the disc, attempted throws, completed throws, etc. To find opponent classifications, the input will be samples of opponent data per game, each of which include the score differential, the opponent's national ranking, the record for games from that year against Stanford, etc. The output in both cases will be cluster centroids, representing the average player or opponent from each cluster, and cluster assignments for each of the player-game or opponent-game samples.

To find features that predict team success, the input will be the opponent cluster number, the tournament number, the round number, and the average performance across the player categories of players who match that specific classification. The output of the predictor will be a simple boolean value that represents whether or not Stanford Women's Ultimate won the game. The output of the wrapper feature selection should be a list of features that contributed the most to error reduction in the classifier.

2 Related Work

While machine learning has been applied in many ways to sports data, data-driven analysis of Ultimate Frisbee has been sparse. In this section, we focus on current applications of data analysis to the sport.

2.1 Ranking Systems

The foundation for most current rankings and predictions in American Ultimate Frisbee is the USA Ultimate national rankings. Since January 2014, a team's national ranking consists of the average of previous game ratings from the year, which consist of a rating differential, calculated based upon the score differential of the game, plus the opponent's previous rating [1]. The rating differential equation is built to weight games so that each goal scored matters more when games are close, the maximum differential is 600 (achieved only when the winning score is more than twice the losing score), and each game decided by one point gets the same differential of 125 [1]. While this algorithm generally provides good rankings, it unevenly weights points scored and struggles with early season teams and teams that do not have a lot of overlapping play, as play is restricted to within region.

While this main ranking system still stands, other systems have been proposed. In 2016, Cody Mills, an analyst for Ultiworld, the main online news source for Ultimate Frisbee, suggested the use of the Elo Rating Algorithm, which is widely used by sports statisticians for leagues such as the NFL and world football, that uses factors of relative ratings, margin of victory, and a weight on recency of the game [2].The following year, Mills suggested the use of probabilistic ranking system to calculate the top twenty teams by taking the USA Ultimate rankings and calculating the collective probability of a team from a region being in the top twenty [3].

These models provide a reasonable, retroactive set of rankings - due to the nature of their creation, they are much less effective with less data earlier on in the season. Even so, the common sentiment among Ultimate coaches is that the rankings do not generally yield strong head-to-head predictions, particularly between teams who have not seen each other before, at the end-of-season tournaments that are well-known for upsets. Unfortunately, there have not been

TABLE 1
Example of Raw Player Data Sample

Year	Tournament	Opponent	Round	Player	Position	Points On	Comp. Throws
2014	Sectionals	Sonoma	Pool Play	Slim	Handler	8	9
Attempt. Throws	**Comp. Hucks**	**Attempt. Hucks**	**Goals**	**Assists**	**Defensive Plays**	**Drops**	
11	2	3	0	0	0	0	

any quantitative studies tracking rankings over the season and game results. In addition, since the models draw upon limited information - game scores and relative rankings only - all they do is provide a general sense as to *which* teams are doing well, while providing little insight into *why* teams are doing well in particular games.

2.2 Predicting

Ultimate Frisbee media sources who look to make predictions often do so with little regard for the rankings, primarily citing past game scores, personal knowledge, and insight. For example, Ultiworld published an article previewing the quarterfinals at the 2017 D1 College National Championships where the only game statistics that were mentioned were previous game scores, a couple player's heights, and the occasional tournament seed [4].While the article is full of other excellent analysis and there are ample scores from the teams facing off at top tournaments all year, there are no stats to quantify a stated weakness against a particular type of defense, or even some metric to compare two all-star players who are both listed as "athletic" or as "top throwers" or as having the same position.

2.3 Applied Machine Learning

In 2015, a student final project at Harvard University looked at American Ultimate Disc League (an Ultimate Frisbee pro league) data that contained similar features to my dataset, and used machine learning to explore differences between players, indicators of Twitter fame, and different predictors of point success [5]. The multi-part project found differences between defensive and offensive players and between fast, clean, and long points through k-means clustering [5].The project was also able to use Random Forests to find that a player's twitter popularity is largely dependent upon the goals they threw, their defensive plays, and the offensive points they played [5].

While my project also aims to analyze player features, I would like to cluster without assumptions and analyze the found player and opponent clusters on a per game basis. I would also like to attempt to build a predictor that analyzes important features across the team, which takes into account the influence of team makeup.

3 Dataset and Features

The player data in the dataset is all the property of Robin Davis, head coach of Stanford Women's Ultimate. The opponent data comes from USA Ultimate, the national governing body of Ultimate Frisbee, from the college tournament and college ranking archives on their website. To build the dataset for this project, I have merged information from both datasets and added some additional, derived features.

Of the player data samples, the training, validation, and test sets consisted of 1791, 403, and 442 samples respectively. Of the opponent data samples, the training, validation, and test sets consisted of 209, 24, and 25 samples respectively. The division of samples depended on the ability to match the player and opponent data for particular games. See Tables 1 and 2 for examples of raw player and opponent data.

Pre-processing was done with the scripting utility of Google Spreadsheets. Categorical labels were converted into numerical fields. Using the raw data, additional features were added, including absolute value of score difference in the game and ranking score as fraction of maximum score in opponent data and percent points played and throw completion rate in player data. Player, opponent name, year, points played, and game score were removed. A set of game indexes was built that numbered each game in the opponent data and matched those numbers to each sample of player game data. Finally, I derived the ground truth, the derived boolean feature of whether Stanford won (1) or lost (0) the game, and linked these to the game indexes. Prior to running any machine learning algorithms, each field was normalized separately across all samples.

For the predictor, I needed a way to combine the samples of player data and opponent data into one sample. I did so by classifying each sample of player data by finding the nearest cluster representative, then averaging the player data for the players in each game who belonged to the same cluster. To hide opponent information that might give away the result of the game, I hid the opponent data fields and solely used the cluster number of the center closets to the opponent data. The result was 103 train, 24 validation, 25 test samples of the following form:

Opponent Type, Tournament, Round, Averaged Stats for Players in Cluster 0, Average Stats for Players in Cluster 1, ...

4 Methods

This project involves three main algorithms that function as a pipeline for the data. First, K-Means is used to cluster the data. Then, logistic regression is used to build a predictor of game success. Finally, feature selection is used to analyze which features impact the accuracy of the predictor the most.

4.1 K-Means

K-Means was used to cluster the opponent and player statistics per game into types of opponents and players.

TABLE 2
Example of Raw Opponent Data Sample

Year	Tournament	Opponent	Stanford Score	Opp. Score	SU Wins	Opp Wins	USAU Ranking
2011	Regionals	UCSB	12	15	1	2	1
Opp. Ranking Score	**SU Ranking Score**	**Total Wins**	**Total Losses**	**Region**			
1819	1770	34	6	SW			

The K-means algorithm is a form of unsupervised learning that looks to find naturally occurring clusters in the data by minimizing the distance between some cluster representative and members of that cluster for each cluster. The algorithm iterates between selecting cluster representatives from the clustered points and assigning data points to the nearest cluster, looping until the cluster assignments no longer change.

For this project, all data was normalized by feature prior to clustering. The mean and standard deviation from the normalization were stored and used to later un-normalize the cluster representatives for analysis. cluster centers were initialized to random samples in the training set. Clusters were found by minimizing the Euclidean distance between the cluster centers and the samples belonging to each cluster.

$$J(c,z) = \sum_{i=1}^{m} ||x^{(i)} - z_{c^{(i)}}||_2^2 \tag{1}$$

New cluster representatives were chosen by averaging the feature values for each of the samples belonging to each cluster. As a result, the cluster representatives were not data points in the dataset, but were derived from the data. Normalized cluster representatives were saved to be used to derive features for the predictor.

4.2 Logistic Regression

Logistic regression was used to make the win/ lose predictions per game.

As discussed in the Dataset section, each sample of the player and opponent data was first classified by being assigned to the closest cluster representative (using Euclidean distance as in equation 1) and then merged using the stored game index.

The logistic regression classifier predicts the label of a given sample by evaluating equation 2, where $h_\theta(x)$ uses the sigmoid function to turn the inner product between our weights, θ and our sample, x, into a prediction between 0 and 1. We classify by rounding that prediction to the nearest value, 0 or 1. In this case, as the ground truth has been defined, 0 would represent a Stanford loss and 1 would represent a Stanford win.

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \tag{2}$$

To solve for the values of θ to use, I used stochastic gradient ascent to maximize the log-likelihood of the data. The log-likelihood equation is defined in equation 3. Maximizing the log-likelihood maximizes the conditional probability that y is of the correct class conditioned on a sample, given our weights, θ. In using this method, we assume that our data varies like a Bernoulli variable and that each sample is independent and identically distributed. While these assumptions are not strictly true, we include factors like tournament number and round to represent some of the dependencies that may exist and predict games independently moving forwards.

$$l(\theta) = \sum_{i=1}^{m} y^{(i)} log h_\theta(x^{(i)}) + (1-y^{(i)}) log(1-h_\theta(x^{(i)})) \tag{3}$$

Gradient ascent works by repeatedly taking small steps in the direction of the gradient of the log-likelihood function. To speed things up, we use stochastic gradient ascent, where we use one sample of data at a time to update θ,as opposed to calculating the gradient over all of our samples. As seen in equation 4, the update to theta combines the gradient, the sample data, and α the learning-rate or size of the step we are taking in the direction of the gradient. When the update to the gradient is very small, we consider the resulting θ as the value that maximizes the log-likelihood, as we have reached a local maximum.

$$\forall j, \theta_j = \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} \tag{4}$$

4.3 Feature Selection

Using the logistic regression model, I ran wrapper or forwards feature selection, based on both the class notes and the algorithms described in ”Integrating Feature Selection Algorithms for Classification and Clustering” [6]. Since feature selection can be slow, logistic regression was ideal as a simple and fast model.

Forwards feature selection works by starting with an empty set of features. We repeatedly test adding each unselected feature to our current set of features by training the logistic regression model and calculating the error on a validation set. After training models for each of these potential sets of features, we then accept the one with that leads to the greatest reduction in error and return to testing adding additional features.

Since I am interested in the features that most reduce error, and to have a reasonable running time, the program institutes a feature cap at 17. The program also caps the logistic regression runtime at 50,000 iterations, under the assumption that further runtime will not lead to significant improvement if the gradient ascent has yet to converge.

5 RESULTS AND DISCUSSION

5.1 Clustering Results

After testing a couple of different number of clusters, I settled on 8 opponent clusters and 16 player clusters. These

TABLE 3
Sample of Player Cluster Centers

#	Position	Throw%	A. Hucks	Assists	%Played	Goals
1	-.98	.92	1.22	1.91	.67	.30
2	-.33	.86	5.23	3.05	.79	.60
3	.64	0.0	0.02	0	.15	.22

TABLE 4
Sample of Opponent Cluster Centers

#	Pt diff	SU Wins	Opp Wins	Ranking	Total Pts	Region
1	11.88	1.13	0.00	129.63	14.12	0.06
2	3.86	.46	1.96	4.54	22.21	1.07
3	5.03	1.07	0.41	10.00	20.69	5.34

numbers were chosen to be slightly larger than the values around which the mean squared error (equation 5) stopped dropping significantly. A slightly larger number of clusters was chosen to maintain more variety with which to analyze features and on which to run feature selection, and produced more stable cluster centers than a smaller number of clusters.

$$\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - h_theta(x^{(i)}))^2 \tag{5}$$

Table 3 shows a portion of the unnormalized cluster representatives from the player clustering. Cluster #1 can be interpreted as representing specialized main throwers who have very high completion rates and don't take many chances (hucks are long throws that are more likely to be turn overs). By comparison, cluster #2 can be interpreted as thrower and receiver hybrids who make more risky decisions (more hucks, lower throwing completion percentage) but tend to play more and throw more assists. Cluster #3 represents players who play very rarely, and who do not complete many passes, but occasionally score goals. With a similar analysis of the other clusters, we can see that we have successfully captured some naturally occurring distinctions in players and playing style in a more nuanced way than the common binary division of thrower and receiver.

Table 4 shows a portion of the unnormalized cluster representatives from the opponent clustering. Cluster #1 represents weaker (low ranking) teams that are from the same region as Stanford, but are often blown out by large margins. Clusters #2 and #3 both represent teams that Stanford plays long, relatively close games with. Cluster # 2 tends to win the games, and they tend to be extremely high ranked and from the historically strong Northwest region. On the other hand, Cluster #3 appears to generally lose the game, is ranked around 10th, and generally comes from regions far away from Stanford.

While the cluster centroids seem to portray several known Ultimate Frisbee player and opponent archetypes, the mean-squared cost of the clustering, which varied in the range 5-8, and the resulting cluster centroids appeared to vary quite a bit between runs (accounting for different potential orderings of clusters). This variance seems to indicate that either we dont have enough data to make stable clusters, we should continue investigating an appropriate number of clusters, or that the data itself does not cluster neatly.

5.2 Prediction Results

When trained with a learning-rate, α, of .001, the algorithm usually converged in under 50,000 iterations. A higher value of α often lead to under/ overflow of the exponential function, and smaller values of α led to very long run times.

TABLE 5
Sample of Opponent Cluster Centers

	Error		Accuracy	
# features	Train	Test	Train	Test
227	.0002	.2656	1.0	.72
17	.0131	.2706	1.0	.64

My main metrics of success were accuracy and error. Accuracy is defined as the fraction of examples in the validation or test set that were correctly classified. For error, I used the mean squared error (equation 5) between the raw predicted value ($h_theta(x)$) and the ground truth labels.

Train and test results for the logistic regression predictor are listed in Table 5.

For the logistic regression model with all the features, a very low train error and a high test error suggests that the model may be overfit to the train data, and that it may require a wider variety of data and more regularization to generalize well. Another indication of this overfit is the fact that while the model was able to attain 100% accuracy on the train set, its accuracy on the test set was only 72%. At the same time, the low train error and 100% accuracy is a good indicator that this model can properly represent the dataset.

After feature selection, we find the same pattern of low train error and a high test error that are indications that there is also high variance in the model, a problem that could potentially be fixed by removing the artificial cut-off at 17 features and adding in more data samples.

Interestingly, the error in the test set for the predictor using all 227 features and the error in the test set for the predictor using just 17 features was already within 0.01, although the difference in accuracy was 0.12 (corresponding to 2 more games classified correctly using all the features versus just 17). This small difference suggests that we don't need many more than the top 17 features to as accurately predict the result of a game. With some more features - or better features - and more samples, we may be able to build a much more accurate predictor.

5.3 Top Feature Analysis

Feature selection yielded a list of features numbers that could be traced back to a particular feature in a player cluster. A portion of the results are shown in Table 6

Looking at the results in the table, we can draw some interesting and logical conclusions on how the performance of different components of a team can impact the prediction as to whether or not they win. The large positive weight

TABLE 6
Sample of Features Selected

Feature	Cluster	Weight
%Played	14	.64
Comp Throws/ Pt	8	1.37
Drops	7	-.91
Comp. Hucks	15	1.45
%Played	15	-.72
Attempt. Throws	2	-.30

on the percentage of the game played by players in cluster 14, which consists of strong throwers who have high completion rates and many assists, suggests that the more these players play, the more likely the team is to win. On the other hand, drops by players in cluster 7, has a high negative weight, which is logical as a drop constitutes a turn over! The completed throws per point by players in cluster 8, which consists of players with very few throws who play in early season tournaments, has a strong positive weight, which suggests that the more the newer players complete passes in early tournaments, the more likely the team is to win.

Other features have less clear interpretations. One interesting set of features the strong positive weight on completed hucks and strong negative weight on the percentage of points played for cluster 15, which consists of receivers in early season games who play a third of the points and contribute a small but equal amount across the other stats. Perhaps when they make the unexpected contribution of completed hucks (long throws), it is strongly positive, but on the other hand, having most of the game played by ”average” players isn’t the most effective winning strategy in college women’s ultimate.

While it would be surprising that the opponent type was not a feature that reduced error, its lack of appearance on the feature list was likely due to the fact that I reduced it to a single category that had no numerical significance - larger numbers or smaller numbers did not necessarily mean anything. I had thought to remove the effects of including features such as how many times Stanford had won or lost against the team in case they were too strong of a predictor, but I now believe that I should have instead included features of the cluster represented that the opponent was closest too or came up with a numerical translation of opponent category that made more sense.

As stated previously, although this subset of selected features did almost as well on the test set as the full set, it still did not perform as accurately as desired. Additional feature selection methods with the removal of the artificial cap at 17, feature creation, especially over clusters created by more samples of data, and using additional feature analysis methods to understand the dimensions of the data (PCA, Mixture of Gaussians, etc.) would likely be helpful in increasing the accuracy of the features returned by feature selection.

6 Conclusion and Future Work

While it was nice to see that Ultimate Frisbee knowledge did reflect itself in the clusters and features, the predictor is still not as accurate as I would have hoped - although there is no current metric for the accuracy of the sportswriters who currently provide predictions with which to compare. The poor test accuracy (worse than the 74% chance of being right if you guessed that the team would win every time) was disappointing albeit unsurprising given the small amount of data and the simplicity of the model, which was chosen to ensure human readability of features. A low training error shows that the model does have promise in its ability to capture the data, but that may change with additional data from teams and games that do not follow the traditional Stanford method of playing. Finally, a small subset of the full feature set was able to almost as accurately predict the test set, which suggests that more feature analysis, to craft better features, and feature selection, to find a better set of features, would be helpful when starting to implement more advanced machine learning techniques.

Appendix A
Ultimate Frisbee Resources

For a quick overview of rules and history, checkout whatisultimate.com.

For information on Stanford Women’s Ultimate, see our website. Here is video footage of Stanford Women’s Ultimate winning the College D1 National Championships in 2016!

For information on USA Ultimate, see their website.

Acknowledgments

Thank you to Robin Davis for sharing her years of data for me to explore and for being an excellent coach and role-model.

Thank you to the staff of CS 229 for the helpful feedback and for teaching the methods used in this project!

References

[1] play.usaultimate.org. (2017) Team rankings. [Online]. Available: https://play.usaultimate.org/teams/events/rankings/#algorithm

[2] C. Mills. (2016, month=may, publisher=) Ranking ultimate teams with the elo algorithm. [Online]. Available: https://ultiworld.com/2016/05/25/ranking-ultimate-teams-elo-rating-algorithm/

[3] ——. (2017, Apr.). [Online]. Available: https://ultiworld.com/2017/04/14/exploring-probability-based-bid-allocation-system/

[4] C. Mills, R. Thompson, T. Wissel, and P. StegeMoeller. (2017, May) D-i college championships 2017: Quarterfinals preview (men’s). [Online]. Available: https://ultiworld.com/2017/05/28/d-college-championships-2017-quarterfinals-preview-mens/

[5] J. Martinez, E. Houlihan, R. Kerr, and K. Hsu. (2015) The ultimate analytics. [Online]. Available: http://karinehsu.github.io/cs109-final-project/website/

[6] H. Liu and L. Yu, “Toward integrating feature selection algorithms for classification and clustering,” *IEEE Transactions on knowledge and data engineering*, vol. 17, pp. 491–502, Apr. 2005. [Online]. Available: http://www.public.asu.edu/~huanliu/papers/tkde05.pdf