Magic: The Gathering

Deck Performance Prediction

Roger Hau, Evan Plotkin, Hung Tran

Introduction:

Magic: The Gathering (MTG, or Magic) is the oldest and most popular trading card game played today, due in part to the complex interplay of thousands of cards. In two-player games, each player constructs a "main" deck (referred to as simply 'deck' hereafter) which consists of 60 cards, allowing players to pursue an enormous number of strategies and card combinations. When experienced Magic players select cards for their decks, they work to select both powerful cards as well as cards that complement each other. In other words, a Magic deck's strength is based not only on cards that are powerful individually, but also are synergistic with the other cards in the deck.

Our project seeks to accurately predict the strength of a deck in tournament play by assigning it a score between 1 and 1000. In order to assign scores to a deck, we have examined several deck scoring algorithms that consider both the individual cards in a deck as well as the synergy within that deck.

Methods:

Overview

The ultimate goal is to train an algorithm that, given a deck of Magic cards, will be able to assign this deck a score that reflects its true tournament performance. To generate the deck scoring, our algorithm considers several features, namely the individual cards that compose the deck as well as various measures of synergy, which we must generate from the training data. Thus, there are three primary steps in generating the deck scoring:

1. Our algorithms must generate a set of feature parameters representing synergy from the training data.
2. Our algorithms must compute a deck scoring using the generated feature parameters and the presence of individual cards.
3. The error, or difference between the predicted and true value, will be calculated and recorded.

Data Sources

To obtain a large number of high-quality decks, we collected tournament results from Wizards.com, a site that hosts internationally ranked, official Magic tournaments. The tournament results included the decks and their respective rating. In total, we have collected 12,500 decks representing 600 cards.

Features

We hypothesize that the individual cards of a deck, along with the synergies within that deck would be strong predictors of a deck's tournament performance. We resolved to evaluate two particular areas of synergy: the presence of small combinations of synergistic cards, and the overall synergy of a 60-card deck. However, there is no simple way for our algorithm to identify complementary card combinations in the context of magic. Similarly, there is no simple way for our algorithm to analyze a deck of cards and determine the archetype. Thus, we must develop computable heuristics to represent card combinations and overall deck synergy. A more in depth discussion of these heuristics and feature generation is below.

Deck Archetypes (K-means Clustering)

Algorithm

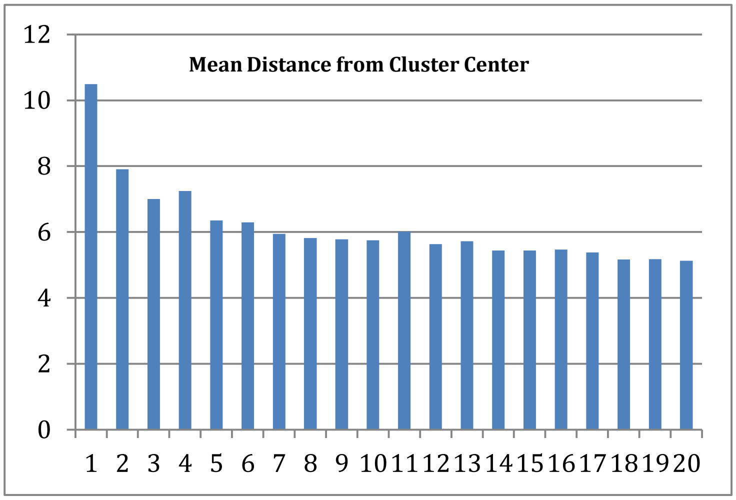
An important characteristic of a magic deck is its archetype. We've mentioned archetypes before, but let us more rigorously define them. In Magic, when two decks have very similar compositions, they belong to the same archetype. Since archetypes have relatively specific composition requirements, archetypes also tend to define the strategies available to a particular deck. Strong archetypes emerge from human analysis and tournament results, and well-performing archetypes often make up the majority of decks in a given tournament. However, although archetypes do have composition requirements, there is still variability among decks of the same archetype.

We hypothesized that archetype is be a strong indicator of tournament performance. As a result, our group sought to capture archetypes as a feature and determine an algorithm to classify decks based on their archetype. Since we sought to determine *groupings* of decks, our initial choice was to use the k-means clustering algorithm to classify our decks.

In our implementation of k-means clustering, our clusters $\mu_1,\mu_2,\ldots,\mu_k \in R^n$ were initialized to be the composition of randomly selected decks from all of the decks in our training data. Since most tournament decks belong to an archetype and the deck compositions are good approximations of a general archetype composition, it is reasonable to initialize the clusters close to their eventual "true" values. Our feature vector x represented a deck, and corresponded to a vector of frequencies, where each index corresponded to a particular card.

Results

To evaluate the quality of our clusters, we found the average distance of a point in the cluster to the cluster center. The smaller the average distance, the better the quality of the cluster. After running our algorithm with varying numbers of initial clusters, we found that the average distance converged with 20 clusters. We also performed a manual inspection of our clusters, which increased our confidence in this algorithm. A personal examination of the top cards in a given cluster revealed that that cluster represented a known popular archetype.



Analysis:

The clustering algorithm worked exceptionally well, and correctly classified the decks into their appropriate archetypes. Although k-means clustering worked well, we wanted to check and see if an alternative algorithm would produce the same results, specifically PCA.

Our PCA algorithm identified deck archetypes as hidden variables that influenced the presence of cards in a deck. A visualization of the clusters verified that the PCA output was indeed correct.

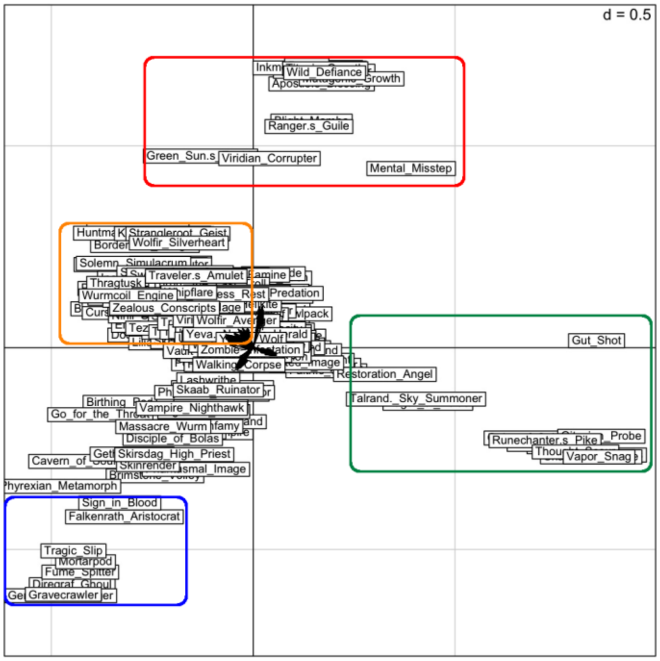


Figure 1: Scatterplot of Data Projected on PCA Components

The below graphic is a visualization of two axis created by our PCA analysis. The circled groups of cards actually represent deck archetypes. The green grouping is a type of "Delver" deck, the red grouping is a type of "Poison" deck, and the blue grouping is a type of "Zombies" deck. These groupings are far from the origin because the cards within them are typically used only in that certain archetype. The orange grouping is interesting, because many of the cards in that grouping are used in numerous archetypes. As such, it makes sense that this grouping is more clumped together and closer to the origin.

When comparing our PCA algorithm to our clustering algorithm, we found that they computed very similar groupings. Decks that were clustered together with our clustering algorithm also had similar compressed PCA feature vectors.

Card Combinations (Association Rule Learning)

Algorithm

Since it would be extremely difficult to evaluate the true synergy of card combinations within the context of Magic, we instead used association rule learning, which identified combinations where constituent cards primarily appeared in conjunction with the other cards in the combination. The number of possible combinations, though, is enormous, and so we set an upper limit on the grouping size to be 4. $Number\ of\ Combinations\ = \sum_{n=1}^{4}\binom{N}{n}$, where n is the size of the grouping, and N is the number of unique cards in the training set.

In order to filter out superfluous combinations, we assigned each combination a rating (which represents the synergy of a combination) using the following functions:

$$Rating \; = \sum_{X \in Powerset(Combination\ C)} Confidence(X\ implies\ C - X)$$

$$Confidence(set\ X\ implies\ set\ Y) \; = \frac{Frequency(X \cup Y)}{Frequency(X)}$$

The confidence of an associative rule is a measure of how accurate we expect that rule to be. By summing the confidence of relationships between different cards within a set, our rating achieves a measure of how related the cards in that set are.

We then selected the varying numbers of the top-rated combinations as features, i.e. the top 50, 75, 100, 125, and 150 combinations. Our results are below:

As with the deck clustering, we manually examined the top-rated combinations and found that they were indeed logical combinations of cards.

Analysis:

When we examined the combinations, we found that most combinations were simply subsets of the most popular archetypes, and suspected that using combinations as a feature may not add more information to our scoring algorithm.

Scoring Algorithm (Regression)

In order to score the decks using our generated features, we needed an algorithm that would be able to compute a continuous score. We decided that regression algorithms fulfilled this requirement.

Since regression is a supervised learning algorithm, we needed to a "score" for each deck before we could perform the regression. We based this score on the deck's placement in its tournament. The first place deck would receive a score of 1000, the last place deck would receive a score of 0, and the decks in between would be assigned scores in between these values. For example, in a 5-person tournament the scores would be (in order of last to first) 0, 250, 500, 750, 1000.

Weighted Linear Regression

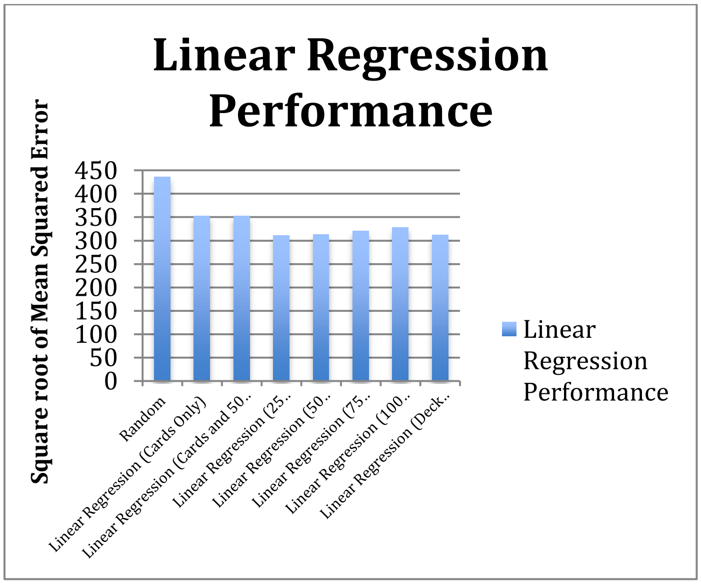
We decided to use weighted linear regression because we believed that the scoring of one deck would be more dependent on the scoring of similar decks, and less dependent on the scoring of less similar decks.

For the input to our linear regression, we considered each unique card to be a feature, the top-rated combinations to be features, and the deck archetype to be a feature.

After implementing weighted linear regression, we found that it was not a good approach for scoring decks. Because many of the decks were so different from each other that they shared few if any cards, the majority of values in the weights matrix were nearly zero. The loss of precision meant that our weights matrix was approximated as a singular matrix, which has no inverse. As a result, Matlab was unable to compute a closed form solution for the weighted linear regression.

Linear Regression

After failing to perform weighted linear regression, we decided to try linear regression without using weights. We performed linear regression using various combinations of the features we discussed above. Below are our results.



Error Calculation

In order to have a data set for testing purposes, we decided to remove 30 tournaments, representing around 1000 decks from our training set. We then used these tournaments as our testing set. We evaluated each deck in our testing set using our linear regression model. We then calculated the mean squared error by finding the average of squared difference between our prediction and the actual score of the deck.

Conclusions and Suggested Future Work

Our goal was to find an accurate scoring algorithm. While our algorithm performs around 25% better than random, there is still a significant error in our algorithm's predicted score. A large issue is high amount of variance within the actual data, which reflects the unpredictability of the card game. Typically, stronger decks only have a marginally higher chance beating a weaker deck, and so a deck's strength does not necessarily guarantee a deck's success in a tournament. For example, it is not uncommon to see a tournament where identical decks achieve significantly different placements.

Although more advanced and more complicated algorithms could possibly improve the scoring accuracy, we have realized that there are fundamental issues in our approach to the deck scoring problem. Specifically, the data we collected does not capture all of the features that would help score a deck, and our generated features captured the same quality. In the rest of the conclusion we will discuss these two issues and make suggestions for future work.

Data

While we had a very large quantity of data points, there are significant factors that contribute to a deck's placement in a tournament that our data does not reflect, such as the skill of the player playing that deck or the properties of the cards. Another important factor is the tournament bracket, which determines which decks will play against each other. Knowing the bracket would allows us to identify which other archetypes an archetype is strong against or weak against. Then, provided with a new, incomplete tournament bracket, a new algorithm might be able to predict a completed tournament bracket instead of calculating absolute deck scorings.

Feature Generation

The two features we generated were the deck clusters and top-rated card combinations. After generating these features, though, we had noticed that the clusters and card combinations appeared to capture the same deck characteristics. After computing the scoring for these characteristics, note that the scoring error using only the clusters, and the scoring using only 50 combinations are 312 and 314 respectively, confirming our suspicions. Even though we had generated two seemingly different features, because the two features represented the same data their combination did not improve the scoring accuracy.

To further improve the scoring accuracy, our team believes that it would be necessary to generate different features. However, our team could not come up with additional meaningful features from this particular data set, which leads our team to conclude that a more informative data set would be required.

References

Tan, Pang-Ning; Michael, Steinbach; Kumar, Vipin (2005). "Chapter 6. Association Analysis: Basic Concepts and Algorithms". *Introduction to Data Mining*. Addison-Wesley. ISBN 0-321-32136-7.

"What's Happening? : Magic Online : Digital Games : Magic: The Gathering." *What's Happening? : Magic Online : Digital Games : Magic: The Gathering*. N.p., n.d. Web. 15 Dec. 2012.