# Predicting the strength of Magic: The Gathering cards from card mechanics

Dustin Fink, Benjamin Pastel, Neil Sapra

# I. INTRODUCTION

Magic: The Gathering is a trading card game, played by millions of players and with over 10,000 unique cards. In addition to products distributed by the game's publisher, Wizards of the Coast, Magic also has a rich secondary market where individual cards are sold. The card values vary significantly, from a few cents to over \$20,000. The price of a Magic card is dependent on its tournament playability, rarity, and collectability. However, the playability of a card is often misjudged when first released and so many cards are valued lower than their eventual worth. Here we develop a discriminative regression model to predict the market price of Magic cards. Our model assesses the value of a card by analyzing only features intrinsic to the card, such as its power, toughness, mana cost, and rules text (descriptions of these card fields can be found in the appendix). This provides a model which not only enables prediction of card value of products released in the future, but also provides insights, through the most relevant features, to what strategies and abilities in Magic are most effective.

## II. RELATED WORK

Previous applications of machine learning to Magic: The Gathering have worked on predicting the worth of cards in terms of price and tournament playability. Pawlicki et. al. used logistic regression and support vector machines to predict price based on price history and tournament playtime [1], while Hau et. al. looked at the synergy of a deck based on the performance of similar decks [2]. However, because both of these projects relied heavily on features that were not part of the cards themselves, they have limited application to new cards which do not yet have price history or tournament playtime. In order to predict the price of new cards, we restrict our feature set to features of the cards themselves.

# **III. DATASET AND FEATURES**

We obtained data for 15,708 unique cards through the MTG JSON database [3]. We removed sets of cards known to have inflated price due to their collectability or scarcity, along with two joke sets. Price of the cards, as of 09 November 2015, was obtained through MTGStocks [4]. As the the range of card prices spans several orders of magnitude, we considered the natural logarithm of the card price as our response variable, rather than the card price alone (Figure 1).



Features were extracted using an in-house feature extractor.

We began with basic numerical card attributes such as the in-game resource cost, power, toughness, and length of rules text, as well as crossings of these features determined by the domain expertise of one of the authors. However, we discovered that the predictive power of these attributes was surprisingly limited, yielding models with only 1-5% improvement on training set RMSE against randomly generated features.

As the strength of a card inherently lies in how it interacts with the gameplay, we eventually developed methods to extract features from the rules text. This part of the card is written in (mostly) natural language and may specify arbitrary effects on the gameplay.



After mixed results with several ad-hoc methods based on gameplay mechanics, we hit an order of magnitude improvement with n-grams. Specifically, we generated a feature for each consecutive sequences of words, numbers, or symbols that appeared anywhere on any card, and gave each card a value of "1" for all sequences present and "0" otherwise [5]. We found that  $n \leq 3$  was sufficient to capture most useful features, while still being computationally feasible.

N-grams provided over 400,000 features, and so a method of feature selection was required to make the model tractable as well as to prevent overfitting. We implemented and compared a variety of feature selection methods, including forward search, filtering features with low Mutual Information, and keeping a large number of features but regularizing them at the model level. Surprisingly, we found that two ad-hoc filtering methods outperformed all the other combinations we tried.

The first filter was to set a threshold r, and remove all features present on fewer than r cards. Low values of rperformed well on training but overfit; empirically, we found r = 200 performed well. However, even with arbitrarily high r, the model continued to overfit on features which differed on only a few cards. For example, "search your library" and "your library for" are common phrases that are present on more than 700 cards; however, there are exactly 7 cards which contain the former but not the latter. So by assigning an extremely large positive theta to the former and a slightly larger negative theta to the latter, the model was able to fit an arbitrary value to those 7 cards, which generalized poorly to the test set. Accordingly, we set a second filter t, such that if two features were identical across all but t cards, one of the features was dropped. The value of t was similarly evaluated and set to t = 200.

# IV. METHODS

After performing feature selection, all cards were converted into feature vectors  $x_1, ..., x_m$ , which formed the rows of the feature matrix X, with an additional intercept term. The log of the price of the cards formed the output vector  $\vec{y}$ . We used the normal equations

$$\theta = (X^T X)^{-1} X^T \vec{y}$$

to obtain a feature vector  $\theta$  that would reduce the root mean squared error (RMSE) between the predicted log-price of the cards  $X\theta$  and their actual log-price  $\vec{y}$ . Thus our final model predicted the price of card *i* as  $h_{\theta}(x_i) = \theta^T x_i$ .

To evaluate our model, we calculated RMSE on an unseen test set. Additionally, to check that our results were not just predicting the average value, we compared our results to the training error of a linear model trained on a random feature matrix.

To investigate whether we should implement a more complex model such as support vector regression, we crudely approximated some of the benefits of support vector regression with a non-linear kernel by crossing all of our features together in several ways. We found no performance improvement or evidence of promising features in these cross features; therefore we decided to focus our resources on feature generation and selection.

# V. OTHER ATTEMPTS

Two additional methods deserve mention that we implemented but ultimately abandonded. Magic players often evaluate cards by comparing similar cards with slightly different attributes or in-game resource cards. We tested two different methods to capture this locality in our model.

First, we tried locally weighted linear regression. We used the normal equations, with euclidean distance between the features and a bandwidth  $\tau$  equal to half the range of the euclidean distances. This method did improve the test RMSE, but the improvement was so minimal we decided it did not justify the increase in training time and complexity.

Second, we conjectured that a model could use the price of similar cards as a reference point, and adjust the price for a new card based on the differences. Therefore, we implemented a method to find the k nearest neighbors of a card and found the average price of those neighbors [6]. Then we trained a model with the differences between a card and its neighbors' features as inputs, and the difference between a card and its neighbors' prices as output. To predict an unseen card, we found the average price of the k nearest neighbors, and adjusted with the model output. This approach was somewhat effective, but could not outperform our original approach, and we decided it did not justify the increase in training time and complexity.

# VI. RESULTS AND DISCUSSION

On a training set of 12916 cards and testing set of 1436 cards we achieved an RMSE of 0.709 log-dollars (\$2.03) on the training set and 0.725 log-dollars (\$2.06) on the test set. For cards over \$1.00, the RMSE was 2.00 log-dollars (\$7.38). This was a 35.84% improvement over a random feature matrix for the training set, and a 34.37% improvement for the test set. We also evaluated the model as a classifier, and calculated its accuracy in terms of evaluating cards above or below a dollar. It had an accuracy of 93%, an improvement over the baseline 84% achieved by randomly guessing based on the prior frequency of being above or below a dollar.



A benefit of using a card-based method was the potential to uncover game mechanics from analysis of the magnitudes of thetas associated with the features. Aside from the "rare" ( $\theta = 1.048$ ) and "mythic rare" ( $\theta = 0.7753$ ) rarities, the model learned which mechanics provide powerful effects in the game such as searching the library ( $\theta = 0.5005$ ) and gaining additional effects from spells and abilities such as drawing additional cards per turn.

### VII. CONCLUSION AND FUTURE WORK

While card prices for Magic: The Gathering are driven by other factors such as market supply and demand and tournament popularity, there is signal in the text of the card itself to help predict price. The usefulness of the n-gram features revealed the importance of card abilities such as searching your library and drawing cards over card power and cost, which are high for some good cards and low for others. It is surprising that numerical features such as power, toughness, and mana-cost were poor signals, even when crossed with each other. Locally-weighted linear regression and nearest neighbors were attempts to compare cards with many of the same features that differed along these lines, but neither of these methods proved very helpful. A more robust crossing of these features with the n-grams may have improved the results, but additional care would be needed to avoid overfitting. While n-grams worked as a proxy for card ability, a robust card parser that codified the rules of the game could drastically improve results. This may prove more feasible for online card games like Hearthstone, where card rules are already codified by the system. Additional avenues for research would include combining this project with previous projects, to determine how these features interact with price history and tournament play. However, the greatest use of this kind of prediction will be figuring out which of the hundreds of new cards may be worth something, and for that, focus on card-specific features is necessary.

### ACKNOWLEDGMENT

The authors would like to acknowledge Sam Corbett-Davies for helpful discussion and advice throughout the project. In addition, we would like to to thank Professor Ng and the rest of the TA team for the fantastic quarter.

### REFERENCES

- Pawlicki et. al. "Prediction of Price Increase for Magic: The Gathering Cards." Stanford University CS229 Projects, 2014.
- [2] Hau et. al. "Prediction of Price Increase for Magic: The Gathering Cards." Stanford University CS229 Projects, 2012.
- [3] Moura, Sergio. MTG JSON. N.p., n.d. Web. Sept. 2015. <www.mtgjson.com >.
- [4] Quantitative Methods, Inc, n.d. Web. Sept. 2015.<www.mtgprice.com >.
- [5] This outperformed variations such as using the number of times a particular sequence appeared as the feature value, disallowing numbers and symbols, etc.
- [6] Finding the k nearest neighbors required significant optimization to be computationally tractable. We eventually found a matrix multiplication approach that was efficient to compute in Matlab.





A Magic card has many fields which describe how the card is played and interacts with the mechanics of the game. At the top of the card is the card's name and mana cost, the amount of resources required to pay for the spell. In the middle frame we find the card's type: land, creature, artifact, enchantment, planeswalker, instant, and sorcery; as well as the sub-type which provides additional description, relevant for certain abilities. The third panel contains the rules text which describes how the card interacts with the game or the abilities it has. In the bottom right hand corner of creature cards is an indicator of the power and toughness of the creature. Lastly, at the bottom information about the card's illustrator is given, as well as copyright information.