Quantifying decision impact in MOBA games

Edward Gan EGAN1@STANFORD.EDU
Justin Huang JTHUANG@STANFORD.EDU
Frederic Ren FREN@STANFORD.EDU

Abstract

In this project we quantify the importance and effectiveness of item-purchase decisions in League of Legends, focusing on the early game. We find that stepwise sequences of classifiers are unable to take advantage of the information provided by early-game item choices in general, suggesting that items as a whole are fairly well balanced. However, a more refined propensity score matching is able to detect a mild but significant effects for specific items.

1. Introduction

MOBA games such as League of legends offer a unique mix of challenges to their players. League of legends (league) in particular is a game between 2 human teams of 5 players. At a strategic level players must choose between different champion (character archetype) and item (equipment) options while at a tactical level players must maneuver around and eliminate the opposing team.

The strategic choices are especially difficult since they must take into account the current game state but have no immediate impact on their own. Players often wonder which of the many strategic decisions they made contributed the most to a win or a loss. In this project we focus on understanding the causal effect different item purchases have on the result of League games.

We examine the relative importance and effectiveness of different item purchases player make. We construct a variety of classifiers which try to predict win or loss based on team champion choices, player gold and experience levels, and player item choice to isolate the effect of item choice. We also use propensity score matching to correct for confounding variables while predicting win rate based on whether or not a certain item was bought.

CS229 Final Project, 2015

2. Related Work

The concept of instrumental variables is useful for understanding the independent impact that item decisions have on a match. In our setting, gold, xp and champion choice are instrumental variables which give us a handle on an underlying game state, since game state is a confounding variable for both match result and item choice.

In (Foster, 1997), the authors discuss how 2-stage least squares (2SLS) can be used to account for confounding variables by first modeling results w.r.t. only the instrumental variables. However, the specific formulas used do not generalize to SVMs, trees, etc... so we combine their high level approach with ideas from forward feature-selection (Guyon & Elisseeff, 2003) to formulate 2SLS-like staged classifier models described in section 4.1.

Within the domain of predicting match results, the authors in (Joseph et al., 2006) used a variety of methods on a high dimensional feature set similar to ours, including decision trees, but did not obtain very high predictive accuracy with any of the methods. Thus, we do not expect very high win/loss accuracy either but that is not necessary for our research question.

In the specific area of analyzing items in league-of-legends, the state of the art metric is raw winrate for each items, as can be seen in third-party apps endorsed by Riot such as (KateOfSpades & Kai). We believe that our methods will be able to give more realistic insight since they will take confounding variable into account.

There is substantial literature on the combination of machine learning and traditional econometric techniques to yield causal estimates. (Athey & Imbens, 2015) discusses empirical methods for combining propensity score matching and machine learning tools such as cross validation to estimate treatment effects, particularly in observational studies with heterogeneous users. (Athey & Mobius, 2012) is an example of an observational study that employs propensity score matching. The authors aim to estimate the causal effect of adding local news content to Google news feeds, where opting into local news is a choice that users make. They build a propensity score model of opting into local news as a function of past browsing behavior

and condition on this information to match users and form a quasi-experiment to test effects on readership.

3. Data

The Riot API (rio) gives us access to a variety of information on a given League match, but does not provide direct access to recent matches, so we began by crawling player-to-player connections to obtain a list of 2,000 North American players in "Silver Tier". All players in silver tier are ranked to have roughly the same (mid-level amateur) skill level. Up to 10 recent matches were pulled for each player for a total of 12,000 matches.

Each match is played between two teams of 5, and each player makes their own item purchase decisions. Since we are modeling individual player choices, we extracted a single sample from each match for a total of 12,000 data points. Extracting more data points from each match would introducing misleading correlations in the data.

We extracted a number of both categorical and numerical features for each match from the json provided by the api:

Category	Example Features
Ally champions	ally_alistar: 1, ally_elise: 0, ...
Opponent champtions	opp_alistar: 0, opp_elise: 1, ...
Items total	boots: 1, longsword: 1, ...
Items @ 10 minutes	boots_10: 1, longsword_10: 0, ...
Gold @ x min	gold_5: 23, gold_10: 250, ...
XP @ x min	xp_5: 43, xp_10: 66, ...
Win / Loss	win: 1

Each team consists of 5 out of 128 possible champions, and similarly each player can purchase any number of distinct items. We encoded the presence or absence of each champion on each team as a binary feature, for a total of 2*2*128 features. Similarly whether or not the player had bought each item by specific points in time is encoded as a binary feature. Cumulative Gold and XP numerical values were measured at 5 minute intervals and each value represented by its own feature.

4. Methods

4.1. Predictive Classifiers

Our method for quantifying the overall impact of item purchases is inspired by both forward-search feature selection and 2-stage least squares modeling. The goal here is to limit the effect of game state, which is a confounding variable since it influences both the final outcome of a match and the items one might choose or be able to buy.

In section 4.1 we limit ourselves to the first 10 minutes of the game to limit variation in game state, and we use champion choices and gold and xp values at 10 minutes as instrumental variables to stand in for the game state. Then, to distinguish the effect of item choices from these, we build three successive models which take into account more features roughly in the order they begin to have an impact

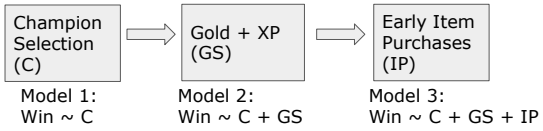

Figure 1. Successive Classifier Models

Figure 1 illustrates how each model takes an additional set of features into account. Model 1 only takes champion choice into account, model 2 adds gold and xp data, while model 3 includes all of the above as well as item choice data. We hope to quantify the overall impact of item choice with the difference in predictive power between model 3 and model 2.

$Impact \sim$ **Model 3 Acc** $-$ **Model 2 Acc**

The specific algorithms we used for each model were l1-regularized logistic regression, l2-regularized linear SVM, and Adaboost decision tree ensembles. The logistic regression minimized the error:

$|w| + C\sum_i^n \log\left[\exp\left(-y_i\left(X_i^T w + c\right)\right) + 1\right]$

Regularized Linear SVM minimizes the error:

$\frac{1}{2}w^T w + C\sum_{i=1}^n \xi_i$

where $y_i(w^T x_i + b) \geq 1 - \xi_i$.

Adaboost trees calculate weights β for decision trees $h_t(x)$ and then classifies based on the sum of the weighted tree decisions:

$\sum_t (\log 1/\beta_t) h_t(x)$

4.2. Propensity Score Matching

A more principled way of isolating causal effects for specific item choices is Propensity Score Matching (PSM). PSM is a method for estimating treatment effects in datasets where assignment to a binary treatment is endogeneous. Our example of this is the purchase of an expensive item (treatment) in League of Legends. Expensive items are frequently only affordable to the team that is ahead, and thus their purchase might be correlated with winning the game even when their actual effect is minor.

The goal of propensity score matching is to estimate the effect of binary treatment X on outcome Y in the presence of endogeneous variables Z. The concern is that Z, which in our case are game state variables, have an effect

on the game outcome Y and influence the item choice X. Then our typical ordinary least squares assumptions would be violated, as the error term ϵ in the estimated model of $Y = \beta_1 * X + \epsilon$ is now correlated with X. PSM handles this endogeneity by modeling the probability of being allocated to the treatment X (buying item) via logistic regression on Z (game state). The logistic regression of X on Z yields propensity scores $P(X = x|Z)$ for each observation. Observations are then paired based on propensity scores via nearest neighbors, such that observations i, j in which $X_i = 1$ are paired with those in which $X_j = 0$ and $P(X_i|Z_i) \approx P(X_j|Z_j)$ and covariates Z are counterbalanced across the entire group. This procedure yields a new dataset where $X \perp Z$, allowing unbiased estimates of the causal effect of X on Y via ordinary least squares regression.

The following provides a (hypothetical) graphical illustration of propensity score matching. Table 1 is hypothetical representative data, where the expensive item Mejai's Soulstealer is more commonly bought by the winning team. Regression of this dataset would assign coefficients which conflate the role of Mejai's Soulstealer on game outcome when gold lead is correlated with both item purchase and winning the game. Table 2 shows what the dataset might look like after the matching process has concluded. We see that both observables and propensity scores are approximately balanced (within some tolerance) between the treatment (Mejai purchased) and control (Mejai not purchased) groups.

Table 1. Representative (Unmatched) Data

ID	Mejai	Gold	$P(X\|Z)$
32	1	+5284	.56
93	1	+2745	.42
420	0	-340	.15
96	0	-3890	.02

Table 2. Propensity Score Matched Data

ID	Mejai	Gold	$P(X\|Z)$
32	1	+5284	.56
742	0	+5350	.57
420	0	-340	.15
278	1	-328	.15

5. Results

5.1. Predictive Classifiers

As described in the methods, we tried to predict match wins in terms of various subsets of the champion, early game state, and early item features.

5.1.1. Linear Models

We started by evaluating the performance of regularized logistic regression and linear SVMS. We evaluated each classifier by its overall accuracy via 5-fold cross-validation. L1 loss yielded the best results for logistic regression while L2 loss worked best for SVMs. We also tried a variety of regularization parameters C ranging from 1e-4 to 1e4, and C=1 was close to optimal.

In figure 2 we compare the accuracy of logistic and linear SVM classifiers for the models described in section 4.1. The two methods perform very comparably. Champion choice in model 1 gives us reasonable predictive power, adding in game state in model 2 helps us significantly, but further adding in early game item choice in model 3 does not improve performance.

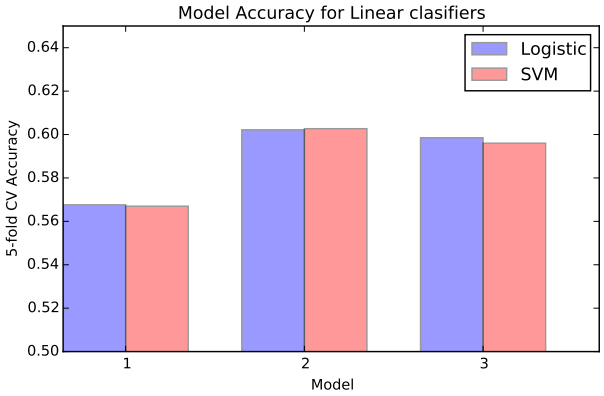

Figure 2. Linear Classifier performance

Table 3 presents the confusion matrix for logistic regression evaluated on a separate test set. The results for linearSVM are very similar and the two methods appear to do slightly better predicting games where the player won.

Table 3. Linear Model Confusion Matrix

	Predicted Win	Predicted Loss
Win	742	464
Loss	506	683

5.1.2. Nonlinear models

Though item choice did not improve predictive performance in linear models, we hoped to see more significant results for nonlinear models which could take into account the situation effectiveness of different items.

In figure 3 we compare the accuracy of SVM using linear, degree-2 polynomial and degree-2 radial basis function kernels, for the models described in section 4.1. The numbers

here are slightly different than in the previous figure since they were obtained from an independent experiment. The linear kernel has the highest accuracy, though it only outperforms the degree-2 RBF slightly (0.01 for model 3). The degree-2 polynomial kernel is significantly worse than the other two. Again, adding game state in model 2 improves accuracy by 0.04 for the linear kernel, further adding item choice in model 3 only improves performance by 0.01.

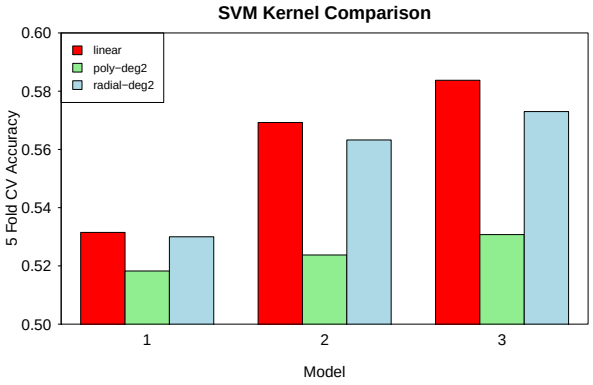

Figure 3. Accuracy of SVM kernels

Along the same vein, we also tried decision trees to see if we could identify context-dependent item choice impact. We evaluated Random Forests, Gradient Boosted decision trees, as well as AdaBoost tree ensembles. For depth d=1,2,3, the AdaBoost algorithm had the best accuracy on 5-way cross validation.

To find the optimal tree depth in figure 4 we compare 5 way CV scores for model 3 ($win \sim champions + state + items$) and see that depth 1 trees perform the best. Increasing the depth yields higher training accuracy but worse test accuracy. For instance, in figure 5 we compare training and test accuracy for depth 3 and see that there is substantial overfitting even in model 1.

The breakdown of the impact provided by the champion choice, state, and items is then given in figure 6. The results are very similar to what we saw for linear classifiers.

5.1.3. Predictive Classifier Analysis

It appears that any additional predictive power provided by item choice is outweighed by the overfitting that occurs. This is especially true for decision trees where any additional depth beyond d=1 yielded worse results. Both decision trees and kernel SVMs were unable to learn any of the truly nonlinear effects of item choice. We tried to reduce dimensionality the by considering only the most frequently purchased 40 items and most frequently played 60 champions but the results did not improve; Principal component analysis was not helpful because each feature only accounts for a small independent fraction of total variance.

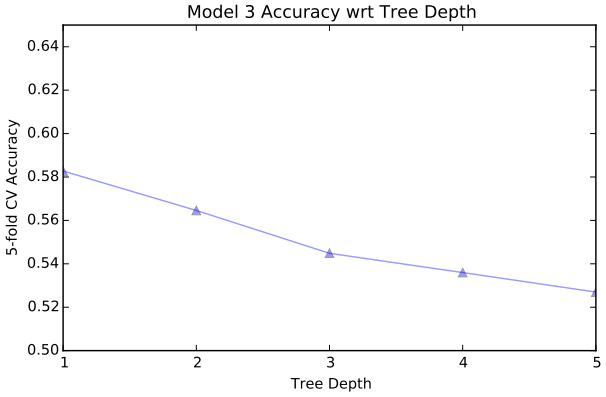

Figure 4. AdaBoost depth parameter tuning

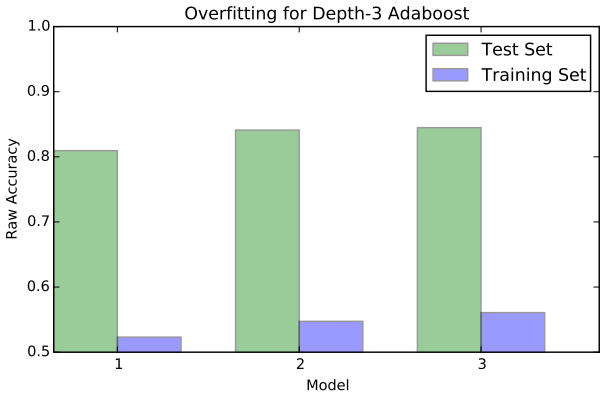

Figure 5. AdaBoost Overfitting at depth 3

5.2. Propensity Score Matching

We implement Propensity Score Matching on our dataset in order to estimate the causal impact of items chosen to be most predictive of winning games via logistic regression. We balance selection into purchasing these items across the observed covariates related to game state: gold5, gold10, xp5, xp10, champion, role, lane, xp differential, damage differential. Logistic regression is used to generate propensity scores and observations are paired based on nearest neighbors using propensity score. The below diagram displays propensity scores both before and after propensity score matching. We see that the distribution of propensity scores is much more similar between treatment and control groups after matching.

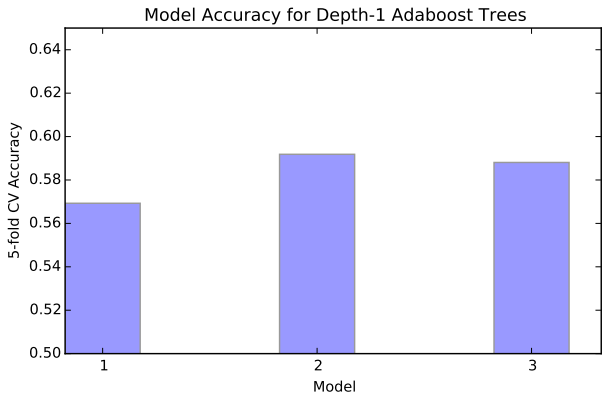

Figure 6. AdaBoost results

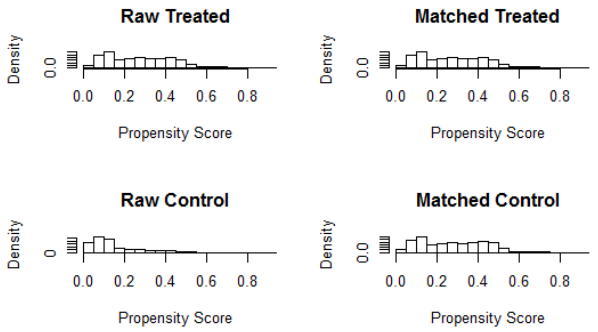

Figure 7. Propensity Scores before and after matching

We then use least squares regression on the matched dataset in order to estimate a causal effect of item purchase on subsequent game outcome. Since this process is relatively time intensive and must be performed with a covariate/item at a time, we decided to look at the most predictive items found via regularized logistic regression and compare those coefficient estimates with the coefficients we find from PSM.

Table 4. PSM vs Logistic Regression Item Coefficients

Item ID (Name)	PSM	Raw
3031 (Infinity Edge)	.06288	.5088
1306 (Enchantment: Alacrity)	.03737	.4407
3706 (Stalker's Blade)	-.0086	.3859
3004 (Manamune)	.00801	.4407
2032 (Hunter's Potion)	.01491	-.4792
1317 (Enchantment: Captain)	-.0645	-.3723

From table 4, we see that items do not always map well from their logistic regression coefficient to their PSM coefficient. While we cannot directly compare the magnitudes, the relative coefficients weightings and sign are noteworthy.

The negative logistic regression coefficient on Hunter's Potion, for example, would suggest that the item is commonly bought by the losing team. On the other hand, the positive PSM coefficient on the item would suggest that in similar situations with comparable champions and game state, players who bought Hunter's Potion fared better. This agrees with general player intuition on the item, as Hunter's Potion is considered a catch-up item that provides good value when behind but lesser value when already winning the game.

6. Conclusions

Predictive classifiers introduced in section 4.1 overfit and perform poorly in taking advantage of item choice data. This suggests that items as a whole are fairly well balanced so that it is hard to distinguish truly powerful items from noise.

Nevertheless, when we focus on specific items, even after correcting for endogenous confounding variables in the game state using propensity score matching, we see that certain items such as the Infinity Edge appear to have a modest but significant impact on the game.

References

Riot games api. https://developer.riotgames.com/api/methods. Accessed: 2015-11-11.

Athey, Susan and Imbens, Guido. Machine learning methods for estimating heterogeneous causal effects. 2015. http://arxiv.org/abs/1504.01132.

Athey, Susan and Mobius, Markus. The impact of news aggregators on internet news consumption: The case of localization, working. Technical report, 2012.

Foster, E.Michael. Instrumental variables for logistic regression: An illustration. *Social Science Research*, 26 (4):487 – 504, 1997. ISSN 0049-089X. doi: http://dx.doi.org/10.1006/ssre.1997.0606.

Guyon, Isabelle and Elisseeff, André. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3: 1157–1182, March 2003. ISSN 1532-4435.

Joseph, A., Fenton, N. E., and Neil, M. Predicting football results using bayesian nets and other machine learning techniques. *Know.-Based Syst.*, 19(7):544–553, November 2006. ISSN 0950-7051.

KateOfSpades and Kai, Praetor. Ripples item analysis. http://maryschmidt.github.io/ripples/. Accessed: 2015-11-11.