Predicting the Outcome of Online Penny Auctions Using Learning Algorithms

Parsana Pillay, Stephen Rebstock, Stevan Clement

December 13, 2012

Abstract

In this report we investigate the application of learning algorithms to online penny auctions for predicting the results of bids on low valued items such as gift cards and voucher bids. We first construct several methods to approach the issue of predicting outcomes of bids and user strategies. We then analyse the effectiveness of these methods using data collected from a popular penny auction website, Quibids. The relation between methods and their performance is discussed.

Problem

A penny auction is a type of auction where each participant pays a fixed price in order to place an incremental bid on an item. After each new bid is placed a timer is reset to a specific amount based on how long the auction has been going. At the expiration of the timer the auction ends and the last participant to bid is awarded the item for the current price. If a auction as been active for a certain time Quibids dose not display the auction to new users. This type of auction can be extremely profitable for the auctioneer as they are paid both the final item fee and the fees from each bid placed in the auction. The incentive to participate in this type of auction is the chance to pay a significantly lower amount than the retail price of an item.

There are a number of penny auction sites on the web. We choose the popular Quibids[1] site to collect our data from. On Quibids each bid costs 0.60 USD and increments the total price by 0.01 USD. The problem we face is how to win an auction using the lowest possible number of bids. In this type of auction the final item price is only a fraction of the total amount the winner must pay, denoted by

$$\texttt{cost} = (\texttt{totalBids} * \texttt{biddingFee}) + \texttt{finalItemCost} + \texttt{additionalCharges}$$

Each of the other participants must pay for the bids they placed, even though they lost the auction and gained nothing.

[1] http://quibids.com/

There are a few other factors for the special case of Quibids auctions. There is a shipping cost of 9.50 USD and a transaction fee of 1.99 USD. The main difference between Quibids and other penny auction sites, such as the former top auction site Swoopo[2], is the buy it now feature included on Quibids. When using this feature you can use the amount you paid for all the bids you have placed in a particular auction to pay the cost of the item. The price Quibids sets for items tends to be higher than in shopping sites or department stores. Additionally, the winner must pay the shipping and transaction fees. Quibids sets limits to the number of auctions you can win in a month and places a cap on the dollar amount of your total winnings. There is also a time limit on claiming your price. Thus a winner of an auction who does not claim their prize within a week loses all the money they spent on the auction.

In this project we try to solve this problem using various methods to predict when an auction is ending and a when winning bid will occur.

Literature Survey

Although online auctions are mentioned extensively in the literature, there are few papers addressing the specific problem of online penny auctions, a particular subset of bidding fee auctions. Bidding fee auctions are a type of English auction, where the price increases and the identity of each bidder is disclosed to all other bidders. [4] looks at the performance of three different auctions. Dutch auctions, where bidding starts at the highest possible price which then decreases until someone bids, at which point the item is sold, a sealed bid auction, where everyone submits a sealed price and the highest price wins, and English auctions. The English Auction, the only auction where people bid against one another in a competitive manner, results in prices at or above the optimal price of the item. The other two auction types result in prices at or below the optimal value.

The earliest mention of the bidding fee auction model occurs in [12] where two bidders are set against each other in order to gain a one dollar bill. They must both pay whatever they bid, and the winner is given one dollar. The article explains how this process ends in a war of attrition resulting in participants making irrational decisions and an overall large profit for the auctioneer. A more recent paper [8] attempts to model penny auctions and declares them to be unpredictable due to high variance of outcomes and unbounded revenue to the seller. In [9] it is shown that although we cannot predict user strategy, as it is too random for game theory, it is possible to predict profit from auctions. User strategy is mentioned again in [3], outlining that bidding strategies of each user involved in the auction affect the result of the auction. There are bidders who act irrationally and aggressive in the auctions and inexperienced bidders who drop out too quickly. The paper finishes by stating that auctions where the majority of users employ sophisticated strategies are far more predictable than those with irrational or novice strategies. [15] remarks that novice users often radically overbid, but retained users often learn quickly and begin to

[2]http://www.swoopo.com/

minimize their losses. A penny auction site's survival then relies on attracting new inexperienced bidders who will lose money. This paper remarks that the key to learning understanding penny auctions is to focus on learning and strategic sophistication done by bidders. However, while focusing on the economics [13] finds that for Quibids, most of the sites earnings come from experienced bidders. Quibids makes money due to the auctioning of its voucher bids, which we plan to study in this paper.

Bidding strategies are again mentioned in [14]. This study shows that 88% of auctions are won bidders whose strategy is to bid persistently throughout the auction. There are two types of persistent bidders: bidders who constantly bid during their time participating in an auction and bidders who alternate between constantly bidding and infrequently bidding as the auction goes on. Auctions usually end on average 30 bids after the last persistent bidder has left or with the last persistent bidder winning. Another important point is that 60% of auctions had the following pattern in the last 10 bids: a single bid at almost 0 seconds and a number of bids immediatly following that single bid.

Other auction sites are discussed in [11]. Specifically, user strategy based on bid timing for Amazon and eBay. As auction experience is gained by a bidder on these sites incremental bidding decreases and last minute bidding, known as auction sniping, increases. This has the effect of changing these sites into sealed bid auctions. If everyone submits a bid last minute, the highest bid will win and it will be impossible to react to another user's bid. Quibids avoids this problem by increasing the auction clock after every bid.

End auction prices on the now bankrupt site Swoopo are discussed in [10] where game theory is used to determine a geometric distribution of penny auction end points that give a specific value for an auction item. Quibids Reports[3], a subscription based service, shows a similar distribution for auction endings. Another interesting in site from this paper is to maximize profits auction sites such as Swoopo maintain a ratio of auctions to currently active users. The ratio is one auction for every thirty three users.

The most cited paper on penny auctions [2] also discusses Swoopo. The paper notes that bidder experience is most important for minimizing expected loss. Despite the extremely lucrative appearance of penny auction for an auction host, it turns out that half of the auctions result in a loss for the auction site, due to the total price of the item plus all revenue from the bids often being less than the cost of the item. However, an auction site still makes an average profit of 50% of the item cost on all items. The paper points out that the end price of an auction is a function of the total number of users and total number of active auctions.

End prices of standard auctions are predicted in [6], and although penny auctions are very different from English auctions, also known as an open ascending price auctions, some of the features used in the analysis for this paper looked promising for penny auctions. Notably, the use of binary classification

[3] http://www.quibidsreport.com/

for eBay[4] auctions could prove to be an excellent tool for our work with penny auctions.

Approach and Rationale

The problem of predicting when to place a bid such that it will be a winning bid is a difficult problem to approach. After reviewing the literature we decided that it may be possible to predict the end with some degree of accuracy above random chance. Collecting the data and manipulating it took the majority of the time as we were unable to locate any free data sets for sites dealing with penny auctions. We wrote our own scraping programs to gather data from the website Quibids Insider[5] because it allowed us to access auctions from the American version of Quibids, featuring far more auctions than the Canadian site, without using a proxy server. Our programs were distributed throughout the undergraduate laboratories and operated using requests mimicking the original site's AJAX functionality to reduce the site load while collecting data.

Once the data had been acquired we manipulated it further to extract and expand features to create data sets for each of our hypotheses. This expansion is discussed below in the Experimental Design. As we were testing to see whether or not we could successful prediction results from penny auction data, a lot of our approach was the tests we used in our experiments.

We picked our hypotheses based on what data we considered useful for winning auctions. The main points covered are: if a next bid will win, if a user will win, and what the end price of the auction will be. If we can successfully predict these points, building a bot to win auctions would be a possible next step.

Plan

Our goal for this project is to show that penny auctions have enough structure that a learning algorithm will be able to make decent predictions on certain aspects of auction outcomes. The following hypotheses were constructed with this in mind.

Hypotheses

1. Given n bids, it is possible to predict whether the next bid will be a winning or losing bid. We are testing whether a bid history can provide a recognizable ending or continuing pattern for an auction.

2. Given a user in an auction, it is possible to predict whether this user will win. Users with specific bidding behaviours are more likely to win

[4] http://www.ebay.com/
[5] http://quibidsinsider.com/

than others. We observed that often auctions end with many bidders still willing to bid believing someone else will bid.

3. Given a particular auction state, it is possible to predict the end price of the auction. This is similar to the first hypothesis, but uses regression instead of classification for more complete information about the final price.

Experimental Design

We used Mlpy [1] to provide a framework for testing our hypotheses. Mlpy is an open source machine learning library written in Python providing functionality for regression, classification, and clustering which allowed us to run a variety of algorithms against our data sets. Making decisions about feature extraction and engineering was going to be an important part of getting decent results. The following features were used for each hypothesis:

1. For our X matrix we populated the row of features with a history of the previous 9 bids. Each bid included information about price at the time of the bid, amount of time left in the auction timer when the bid was placed, the time and date of the bid (auction sites are more popular at certain points in the day), the hour of the bid, the value of the item (we used only voucher bids and gift cards, so we could easily tell the value based on the item's name), whether or not it is a gameplay (some auctions have a gameplay feature to allow the auction winner to win extra bids), whether or not the item is a voucher bid, the number of bids in the last {5, 1, 1/6} minutes, and the time since the last bid. Our Y matrix was a binary classification matrix of whether or not the bid after those 9 bids won the auction.

2. For our X matrix we populated the row of features with the total bids a user had made in an auction, the value of the item, the hour of the auction, whether or not it is a gameplay, whether or not the item is a voucher bid, the total bids a user had made on all auctions in our database, when the user started bidding in an auction, when the user had finished bidding in an auction, and their average time between bids. Our Y matrix was a binary classification matrix of whether or not a user won the auction.

3. Using the same features as the first hypothesis we created an X matrix of a bid history of 9 bids. Our Y matrix was a matrix of final auction prices.

We first created a database of all our auction information. We had collected 30240 bids spanning 2421 auctions with 4551 different users. Each of our test matrices included 2000 random examples from this dataset.

Our first two hypotheses were tested using the following included functions for classification:

- **Large Linear Classification** allows use of the LIBLINEAR library's [5] different solvers. We also use the weight functionality to assign higher weights to winning bids in an attempt to separate their importance from non-winning bids.
- **k-means** uses iterative partitioning to separate data into k clusters.
- **Classification Trees** are a type of decision tree learning based on predicting a class to which a piece of data belongs. This works by splitting the data into a tree model based on recursive partitioning.
- **Kernel Fisher Discriminant Classification** was the only kernel method we used. It is a kernelized version of linear discriminant analysis. Both Polynomial kernels and Gaussian kernels were tested.
- **Maximum Likelihood Classification** calculates a Bayesian probability function which judges how likely a piece of data is to belong to a certain class.

On the third hypothesis we used the following tests for regression:

- **Ordinary Least Squares** minimizes the sum of the squares of the errors to fit a hyperplane to data.
- **Ridge Regression** is like least squares except it penalizes the size of regression coefficients.
- **Partial Least Squares** is similar to least squares except it is better against noisy data sets as the data is first decomposed into a lower dimensional space.

Results

The following Linear Classifications models resulted in no meaningful results for both the first and second hypotheses:

- L2-regularized logistic regression (primal)
- L2-regularized L2-loss support vector classification (dual)
- L2-regularized L2-loss support vector classification (primal)
- L2-regularized L1-loss support vector classification (dual)
- multi-class support vector classification by Crammer and Singer
- L2-regularized logistic regression (dual)

L1 fared much better for the first two.

First Hypothesis

We iterated through a number of combinations of weights and betas for the L1 regularization algorithms. The weights are penalties attached to predicting a bid is a winner when it is not, for all other case the penalty is 1. The below table shows the best results for test and training accuracy based of the geometric mean. Unlike the other hypothesis for this one we used 10000 entries from the entire set of data. 6000 training and 4000 test entries.

Error Type	Weight	beta	Accuracy predicting winning bid	Accuracy predicting non winning bid	Geometric mean of Accuracy
l1r_l2loss_svc Train	0.8	0.3	0.993007	0.997951	0.995476
l1r_l2loss_svc Test	0.4	0.3	0.9875	0.998214	0.992843
Classification Tree Test	N/A	N/A	0.907216	0.99795	0.951502
Classification Tree Train	N/A	N/A	1	1	1

Figure 1 (see Figures section at end of document) shows the training accuracy as we modified the weight and beta for l1r_l2loss_svc. **Figure 2** shows the test accuracy as we modified the weight and beta for l1r_l2loss_svc.

Second Hypothesis

Like in the first hypothesis we iterated through a number of combinations of weights and betas for the L1 regularization algorithms. The weights are penalty attached to predicting a bid is a winner when it is not, for all other case the penalty is 1. The below table shows the best results for test and train accuracy based of the geometric mean.

Error type	Weight	beta	Accuracy predicting winning bid	Accuracy predicting non winning bid	Geometric mean of Accuracy
l1r_lr Train	0.4	0.7	0.662162	0.865079	0.756851
l1r_lr Test	0.8	0.6	0.835655	0.673432	0.750171
l1r_l2loss_svc Train	0.6	0.4	0.810811	0.704762	0.755929
l1r_l2loss_svc Test	0.6	0.4	0.788301	0.723247	0.755074
k-Nearest Neighbour Test	N/A	N/A	-0.30919	0.398524	N/A
k-Nearest Neighbour Train	N/A	N/A	-0.36757	0.612698	N/A
Classification Tree Test	N/A	N/A	0.545961	0.761993	0.644995
Classification Tree Train	N/A	N/A	1	1	1

Figure 3 shows the training accuracy as we modified the weight and beta for l1r_lr. **Figure 4** shows the test accuracy as we modified the weight and beta for l1r_lr. **Figure 5** shows the training accuracy as we modified the weight and beta for l1r_lr. **Figure 6** shows the test accuracy as we modified the weight and beta for l1r_lr.

Third Hypothesis

We found no correlation between the feature vectors we generated and their corresponding prices. Regression testing, ran with the Weka machine learning engine [7], generated "Correlation Coefficients" ranging from 0.005-0.007. This means that there is virtually no correlation between final price and current bidding patterns.

Evaluation

First Hypothesis

L1-regularized L2-loss support vector classification perform very well. We can predict with excellent accuracy the ending of an auction. The main reason for this is the addition of information from the previous 9 bids. Once those features were added our accuracy increased dramatically. It is important to understand this prediction is not enough to win an auction and may even be useless in winning auctions. Thus knowing when an auction is going to end gives one a good entry point into said auction, but then the state of the auction changes since there is a new player and the ending condition could be different or you may get into a bidding war with the expected winner of said auction.

Second Hypothesis

L1 regularization seems to be best, however, it only correct three in four times. This accuracy rate is far too low to be used to decided when the spend $0.60 to bid on an item. Figures 5 to 8 show the results of the L1 regularization algorithms using all tested variations. Lastly this hypothesis was tested and trained on a small subset of all auctions. These auctions had a ending price of less than $1.25 and tended to be the least random.

We tried a lot of different techniques to predict the winning user with a high accuracy, however, the best strategy going forward would be to compile a data set of users over a long period of time and predict a user's strategy and exit point from an auction based on the users previous actions. This would be an excellent way to improve the feature set. The feature set we used clearly does not provide enough information to very accurately predict a auction.

Another idea, may be to use probability prediction to try and predict the user. In the end I believe that predicting a winning user is very difficult because from our observations and that of the papers we read the only signal found to show the ending of a auction is the bidding pattern of the last few bids.

Third Hypothesis

Nothing worked here. One thing we did not try which may increase our correlation coefficients is instead of predicting a price we would predict a price range. Increasing our feature set may also help with this. Knowing how many other auctions are currently active and how many active users there are could affect the end price. As well, using the day of the week, the day of the month, and season may also help more accurately predict the price. However, there is a lot of statistical data showing correlation between item value and auction end price, but the distribution is very wide and may be impossible to predict.

Conclusion: Lessons Learned

Over the course of the project a lot of knowledge was gained about the problem domain. We were able to predict if a bit is going to be a winning bid given our data. However, all three of us still believe it would not be advisable to bid on a penny auction due to the illogical bidding nature of many of the users and the fact the winning many times becomes a war of attrition. We may also be risk averse students who are not willing to let $60 ride to see if your predictions will improve our odds in the real world.

We learned a lot about scraping web sites for content. We initially ran into some trouble with placing too much load on the site we were scraping and were contacted by its web hosting company. After that we rewrote our code to reduce and distribute the amount of requests we were making and had no further problems gathering our data. We also learned the difference between using double or single quotes when passing strings through the linux shell. That error resulted in our data having a far larger number of voucher bid auctions than gift cards.

We had some ideas for further work on the topic. It would be interesting to extend upon our project using the information we gathered to create a classification algorithm. This would allow us to classifying different users to decide whether it is worth it to bid against the remaining bidders.

We didn't have a chance to try our predictions in the wild due to time and money constraints, but this would have been fun. All in all penny auction sites seem to be more "shopping entertainment" sites then actual auctions.

References

[1] Davide Albanese, Roberto Visintainer, Stefano Merler, Samantha Riccadonna, Giuseppe Jurman, and Cesare Furlanello. mlpy: Machine learning python. *CoRR*, abs/1202.6548, 2012.

[2] Ned Augenblick. Consumer and producer behavior in the market for penny auctions: A theoretical and empirical analysis, 2009.

[3] Michael Caldara. Bidding behavior in pay-to-bid auctions, 2012.

[4] Vicki M Coppinger, Vernon L. Smith, and Jon A Titus. Incentives and behavior in english, dutch and sealed-bid auctions. *Economic Inquiry*, 18(1):1–22, 1980.

[5] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[6] Rayid Ghani and Hillery Simmons. Predicting the end-price of online auctions. In *In the Proceedings of the International Workshop on Data Mining and Adaptive Modelling Methods for Economics and Management*, 2004.

[7] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[8] Toomas Hinnosaar. Penny auctions are unpredictable, 2010.

[9] Sonal Mittal. Equilibrium analysis of generalized penny auctions, 2010.

[10] Jeff Nanney. "entertainment shopping" an analysis of profit and strategy in a new auction format, 2010.

[11] Axel Ockenfels and Alvin E. Roth. The timing of bids in internet auctions: Market design, bidder behavior, and artificial agents, 2002.

[12] Martin Shubik. The dollar auction game: A paradox in noncooperative behavior and escalation. *Journal of Conflict Resolution*, 15(1):109–111, 1971.

[13] Eric Stix. An empirical study of online penny auctions, 2012.

[14] Jeffrey Stix. Designing a bidding algorithm for online penny auctions, 2012.

[15] Z. Wang and M. Xu. Learning and strategic sophistication in games: The case of penny auctions on the internet, Feb. 2012.

Figures

First Hypothesis

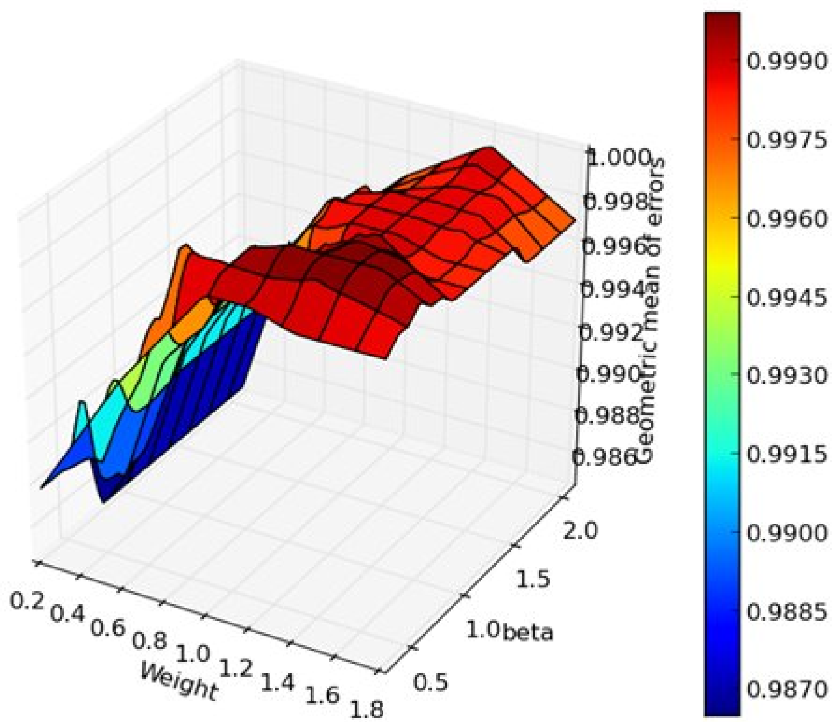

Figure 1 Regularized L2-loss support vector classification Train error

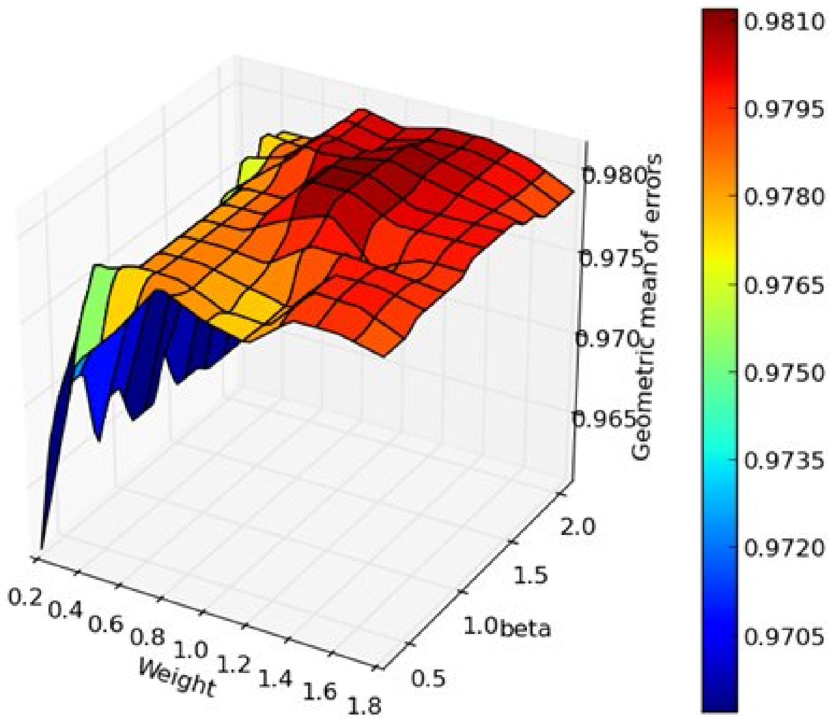

Figure 2 Regularized L2-loss support vector classification Test error

Second Hypothesis

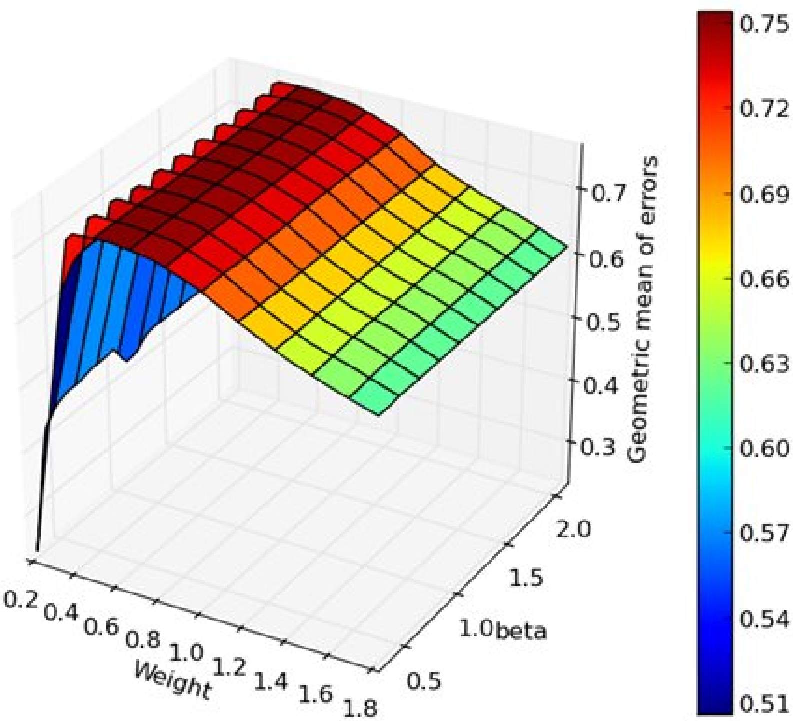

Figure 3 L1-regularized logistic regression Training accuracy

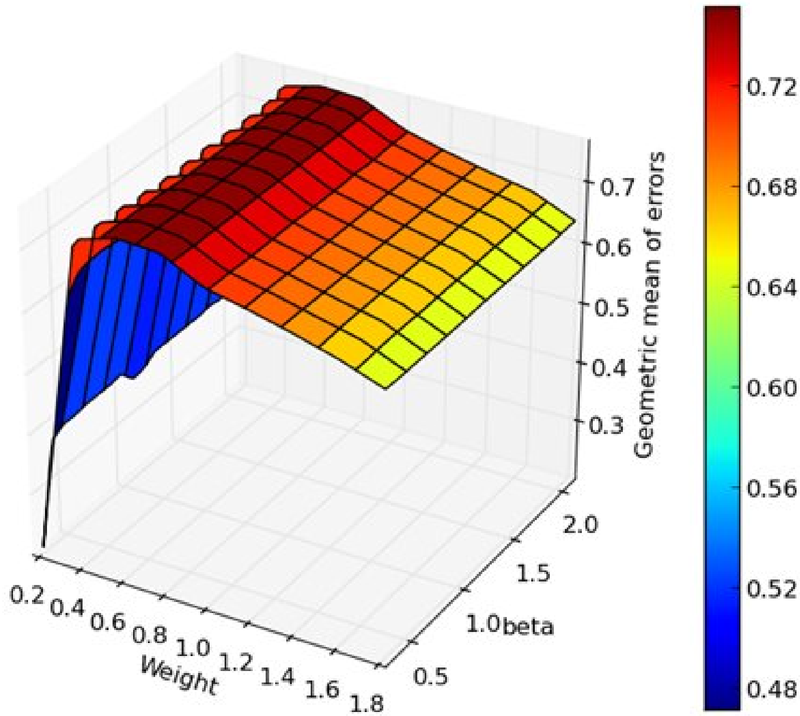

Figure 4 L1-regularized logistic regression Test accuracy

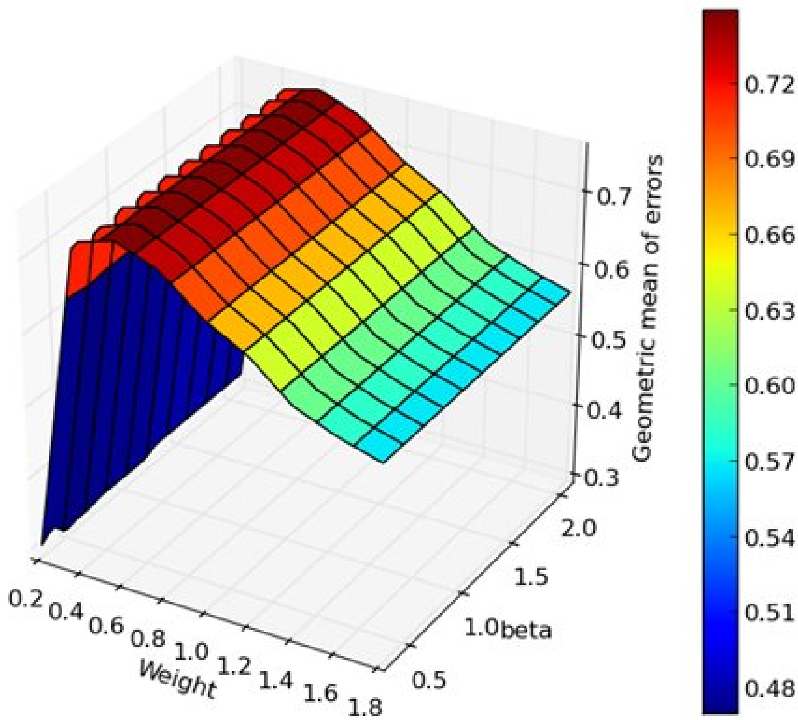

Figure 5 L1-regularized L2-loss support vector classification Train accuracy

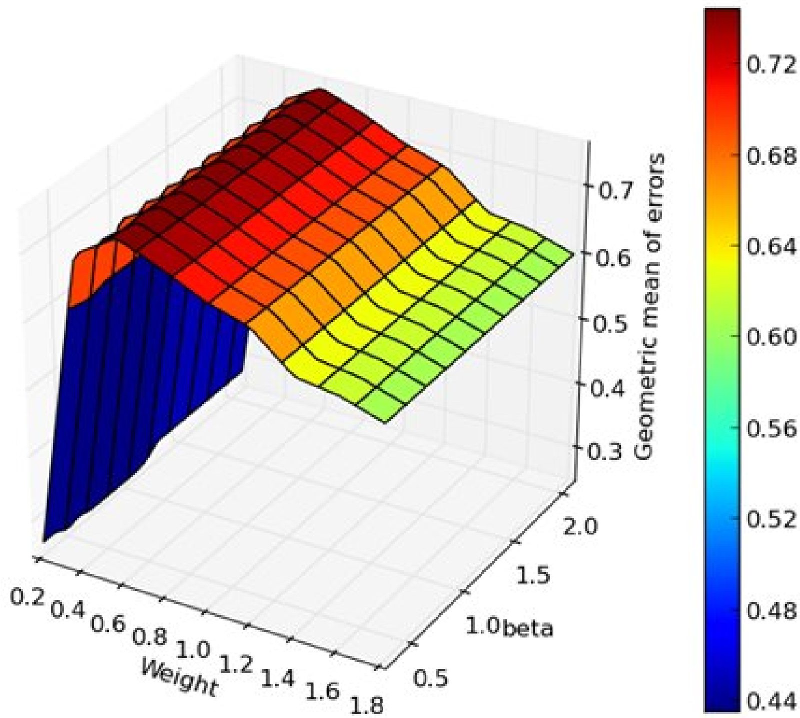

Figure 6 L1-regularized L2-loss support vector classification Test accuracy