Final Program Report: SAMSI Computational Advertising Program Summer 2012

The two-week SAMSI program on Computational Advertising took place August 6–17, 2012. The first four days of the program consisted of technical presentations, a poster session and a discussion of four datasets obtained from the Yahoo! Webscope. Thereafter, four working groups formed to analyze the data sets.

Organizers: Deepak Agarwal (LinkedIn) and Diane Lambert (Google)
Local scientific coordinator: David Banks (Duke University)
SAMSI directorate liaison: Ilse Ipsen (North Carolina State University)

Contents

1 Objectives

Many organizations, including companies, charities, universities, and political campaigns, now allocate a significant fraction of their marketing and outreach budgets to online advertising. Providers such as Google, Yahoo! and MSN (The Microsoft Network) have both enabled this trend and responded to it by constructing new business models. This has led to an exponential growth in online advertising, which in 2010 was estimated to be a multi-billion market worldwide. These business models are the economic engine that is driving the growth of content and functionality on the World Wide Web.

The commercial success of these new business models is based upon mathematical and statistical research that has produced better algorithms for page ranking, recommender systems, auctions, and demand forecasting. The result is a highly scalable system serving billions of ads to users on a daily basis at low marginal cost. This collection of methods and problems has produced a new research discipline called *Computational Advertising* (CA).

The goal of the SAMSI CA summer program was to engage a larger segment of the statistical community in the problems associated with on-line marketing. This is big data science—search engines receive billions of queries each day, cookies aggregate information on hundreds of millions of users, who may be shown display ads from ad pools that number in the hundreds of thousands. As such, statisticians have played a smaller role than is needed. For example, we tend not to have the computer science skills needed to program in Hadoop, a necessity for manipulating large data sets (Google alone processes about 24 petabytes per day). And traditional algorithms for clustering (e.g., as in market segmentation), classification (e.g., as is needed in recommender systems), and robust regression (e.g., as used in predicting click through rates) simply do not scale adequately in such environments.

The SAMSI CA summer program was designed to bring together the relatively small number of statisticians who work in the CA industry with academic researchers, especially new researchers and graduates students, in order to jointly identify novel and practical problems. The ultimate intent is that more statisticians will invest in acquiring the computational tools and algorithmic knowledge needed to effectively contribute statistical expertise in this field.

Summary: Computational Advertising is an important new field that poses challenges at the intersection of statistics, marketing, and computer science. This SAMSI CA program attempts to bring larger numbers of statisticians, especially new researchers, into this domain.

2 Summary of the Technical Presentations

Listed below are the talks, together with a short summary for each, as well as the main points of the discussions after each talk.

Monday, August 26, 2012. *Blake McShane* (Kellogg School of Business, Northwestern University) spoke on "Marketing Perspectives on Online Advertising." He reviewed recent and notable findings from the marketing literature on on-line advertising. In particular, his talk introduced the vocabulary and key ideas that were germane to the subsequent presentations.

Art Owen (Department of Statistics, Stanford University) spoke on "Bootstrapping R-Fold Tensor Data." He described a procedure for approximating confidence intervals in the context of r-factor ANOVA models, where exact solution is provably impossible.

The subsequent discussion identified this contribution as critical enabling technology for quantifying the impact and success of recommender systems. The methodology also applies in network modeling, which is emerging as a key tool for identifying potential customers based on their social networks.

Dean Eckles (Facebook) spoke on "Influence and Homophily in Online Behavior." This talk drew upon ideas from Art Owen's presentation, and also raised the question of how influence propagates through social networks (this relates to recent controversy in the published literature regarding the difficulty of distinguishing causation from homophily in social networks using observational studies). Eckles explained that both peer effects and homophily have practical importance for social marketing. Estimating the role of each, including accounting for heterogeneous effects of different influence attempts, is the goal of some recent direct and indirect experiments conducted at Facebook.

The presentation elicited further discussion on how sequential observation might resolve the identifiability problem.

Justin Rao (Microsoft) described issues in "Measuring Advertising Effectiveness in the Digital Age." Since online advertising offers enhanced measurement of advertising exposure and purchasing behavior, compared to previous vehicles, there is unprecedented potential for understanding the influence of ad campaigns. New data and experimentation platforms allow firms and researchers to measure the true causal effect(s) that advertising has on purchase decisions.

The subsequent discussion focused on definitions for impact, partitioning responsibility among multiple causal factors (e.g., television exposure and Facebook buzz about a product, followed by an on-line display ad, is different from simply presenting the ad to a naive viewer).

Mark Lowe (MaxPoint Interactive) spoke on "Current Challenges in Online Advertising." In particular, he described the MaxPoint business model, in which spatial information on "digital zip codes" is used to select and direct ads to specific geographic communities. The success of such campaigns is measured by lift in sales at local brick-and-mortar outlets, rather than on-line sales.

The subsequent discussion addressed the possibility of using conditional autoregressive models to capture spatio-temporal effects, and how covariate information might be accessed.

Tuesday, August 7, 2012. *Chong Wang* (Carnegie Mellon University) spoke on "Recommendation with a Reason: Collaborative Topic Modeling for Recommending Scientific Articles." The talk describes how modern latent Dirichlet allocation models for topic inference can improve recommender systems. The models considered include the standard allocation model and the nested hierarchical model.

The subsequent discussion considered how this technique might be applied to more general recommendation systems, and how it might take proper account of information from citations and from citation networks.

Mark Handcock (Department of Statistics, University of California, Los Angeles) reviewed "Statistical Modeling of Social Networks" and pointed to strategies for using these models for CA purposes. In particular, he emphasized two approaches: the exponential random graph model, and the latent space model.

During the discussion, there was interest in how betweenness measures might inform prediction for viral advertising. There was also a view that the latent space models had more potential in CA applications than the exponential random graph family.

Claudia Perlich (Media6Degrees) spoke on "Bid Optimizing and Inventory Scoring in Targeted Online Advertising." She described the operation of real-time bidding exchanges, which requires both rapid calculation and game-theoretic strategy. Machine learning techniques are employed in order to learn from past outcomes and thus optimize future bids.

Much of the following discussion focused upon how repeated bidding contests enabled adaptation beyond what was possible in traditional one-off auction theory.

Neel Sundaresan (Director of Research, eBay) spoke on "Paid Search or Free Ads?

Where Search, Recommendations and Ads Co-Exist." The talk emphasized the difference between commerce search engines and web search engines; in particular, commerce search engines often find no matches to a particular query. In the context of eBay, Sundaresan described methods for maintaining user engagement through the interplay of search, recommendation, and advertising, as developed through sophisticated behavioral experiments and format revision.

The subsequent discussion considered how topic models might inform efforts to maintain engagement, and how better experiments might be devised in the light of related work by previous speakers.

Matt Taddy (Booth School of Business, University of Chicago) addressed "Inverse Regression for Analysis of Sentiment in Text." Sentiment analysis is related to topic modeling, but more directly pertinent to CA. Specifically, it can identify how a prospective customer feels about a product, on several different dimensions of sentiment or perception. Inverse regression enables calibration of phrase counts against document characteristics, where the document characteristics have low-dimensional representations. From a technical standpoint, there are issues of non-concave penalized likelihood estimation. Taddy gave examples of several practical applications.

Most of the subsequent discussion related to the examples, although some time was also spent on partial least squares and methods related to Dennis Cook's methods for sufficient dimension reduction.

Hongxia Yang (IBM) talked about "Multi-Relational Learning via Hierarchical Nonparametric Bayesian Collective Matrix Factorization." She explained how this technique can improve inference when data arise from multiple sources that are linked through some network—-such as happens in CA, where the probability of online purchase may depend upon the presentation of multiple ad impressions through different media, and various kinds of information are available upon users. Yang implemented a nonparametric model that employed a hybrid Gibbs sampling algorithm to enable fast computation in the context of article-word co-occurrence features.

The discussion of her presentation focused upon detailing its relationship to CA, and how this method might usefully employ information on someone's local browsing network to improve ad selection.

Wednesday, August 8, 2012. *Nalini Ravishanker* (Department of Statistics, University of Connecticut) spoke on "Fast Computational Approaches for Predictive Inference for Time Correlated Data Streams." She outlined spectral-based clustering and independent component analysis for linear and nonlinear time series, and showed how inference could be achieved through variational Bayes methods. The methodology may apply to inference about online advertising campaign success over time, and changing interest in news reports.

During the discussion, audience members suggested that her methodology could inform forecasts that support advertising contract pricing as well.

Xiaoming Huo (Department of Industrial and Systems Engineering, Georgia Tech) spoke on "Statistical Models in Keyword Bidding." Using data on historical bids and classic auction theory, he described strategies for generating new bids that maximize expected utility.

The discussion addressed the relative value of parametric and nonparametric techniques, and the robustness of the assumption that bidders are rational agents.

Robert Bell (AT&T) spoke on "Regularization for Matrix Factorization." This talk described the theoretical underpinning upon which the predictive ranking algorithm that won the Netflix Prize was based.

Subsequent discussion focused upon the role of L_1 versus L_2 minimization, the impact of sparsity in the rankings, and how to model non-ranked items.

Tian Zheng (Department of Statistics, Columbia University) spoke on "Social Network Analysis Through Randomly Sampled Respondents." Her work addressed sampling issues for incompletely observed networks, using either ego-nominated respondents or aggregated relational data.

Subsequent discussion focused on the latter technique, especially in terms of privacy protection issues.

Christian Posse (Principal Data Scientist, LinkedIn) described "Key Lessons Learned Building Recommender Systems for Large-Scale Social Networks." His talk described challenges that arise when using observed social networks to infer homophily relevant to product recommendations.

The audience was deeply engaged with technical issues that arose, and the discussion tried to link topic modeling methods with network modeling in productive ways.

Liang Zhang (LinkedIn) spoke on "Webpage Personalization and User Profiling." He reviewed research to infer personal traits from user behavior, thereby enabling improved click-through rates and conversion. Methods included bilinear models and parallel matrix factorization, and variations that scaled to billions of operations in the context of "cold-start" campaigns. Discussion focused on the experiments used at Yahoo! to measure the market lift obtained from personalization estimates.

Thursday, August 9, 2012. *David Banks* (Duke University) moderated a group discussion of the key ideas that had surfaced during the presentations, and the group identified areas where fresh research seemed possible and valuable.

Yang Yang (LinkedIn) discussed four data sets, described in Section 4, which had been made available for analysis to the working groups.

3 Poster Session

The presentations on Monday were followed by a poster session with 12 posters, listed below.

1. Tim Au (graduate student, Duke University and MaxPoint Interactive), "Predicting Price Distribution on Real-Time Exchanges."

2. Douglas Galagate (graduate student, University of Maryland), "Using Propensity Score Matching on Click-Stream Data."

3. Mitchel Gorecki (undergraduate, Duke University and MaxPoint Interactive), "Geo-Spatial Modeling of Online Responses at the Neighborhood Level."

4. Genady Grabarnik (graduate student, St. Johns University), "Nonlinear Bid Optimization in Display Advertising."

5. Zainab Jamal (Hewlett-Packard), "Remixing the Marketing Mix: Accounting for Dynamic Cross-Channel Spillover Effects."

6. Alessandro La Torraca (graduate student, University of Milan), "Inferring Demographics from User Navigation Data."

7. Min Li (graduate student, California State University, Sacramento), "Predicting Click-Through Rate for Sponsored Search Advertising on a Chinese E-Commerce Website."

8. Tuwaner Lamar (professor, Morehouse College), "Analysis of a 2nd Order Differential Equation with Lidstone Boundary Conditions—The Existence of Positive Solutions."

9. Alan Lenarcic (postdoctoral fellow, University of North Carolina–Chapel Hill), "Multimembership Social Clustering of Networks for Cuda."

10. David Rosenberg (SenseNetworks), "Leveraging Location History for Mobile Ad Targeting."

11. Jacopo Soriano (graduate student, Duke University and MaxPoint Interactive), "Text Mining in Computational Advertisement."

12. Sarah Tooth (graduate student, Rice University), "Series in Artificial Time."

4 Data Sets

The following four data sets were obtained from the Yahoo! Webscope.

1. Yahoo! Front Page Today Module User-Click-Log Data (1.1GB)

   Online content recommendation represents an important example of interactive machine learning problems that require an efficient tradeoff between exploration and exploitation. Such problems, often formulated as various types of multi-armed bandits, have received extensive research in the machine learning and statistics literature.

   Yahoo! Today module user-click-log data is the first set of benchmark data with this inherent interactive nature. The dataset contains 45,811,883 user click logs of news articles displayed in the Featured Tab (Hero Slot) of the Today Module on Yahoo! Front Page (http://www.yahoo.com). This dataset contains 10 files, corresponding to the first 10 days in May 2009. The articles were chosen uniformly at random from a handpicked pool of high-quality articles, which allows one to use a recently developed method [1, 2, 3] to obtain an unbiased evaluation of an arbitrary bandit algorithm.

2. Yahoo! Music User Ratings of Musical Artists (423MB)

   This Yahoo! Music data represents a sample of Yahoo! user ratings of musical artists, gathered over one month prior to March 2004. The dataset contains 11,557,943 ratings of 98,211 musical artists given by 1,948,882 anonymous Yahoo! Music users. The data set can be used to validate recommender systems or collaborative filtering algorithms, or serve as a test bed for matrix and graph algorithms including PCA and clustering algorithms [4, 5]. Similar topics have been explored in KDD Cup 2011: Recommending Music in Yahoo!

3. Yahoo! Search Marketing Advertiser Bid-Impression-Click Data on Competing Keywords (845MB)

   This dataset contains a small sample of advertiser's bid and revenue information over a period of 4 months. Bid and revenue information is aggregated with a granularity of a day over advertiser account id, key phrase and rank. Apart from bid and revenue, impressions and clicks information is also included. The primary key of the data is a combination of fields date, account id, rank and key phrase. Average bid, impressions and clicks information is aggregated over the primary key. Advertiser account id and key phrase are anonymized.

4. Yahoo! Search Marketing Advertiser Bidding Data (81MB)

   Yahoo! Search Marketing operates Yahoo!'s auction-based platform for selling advertising space next to Yahoo! Search results. Advertisers bid for the right to appear alongside the results of particular search queries. An advertiser's bid is the price the advertiser is willing to pay whenever a user actually clicks on their ad. Yahoo! Search Marketing auctions are continuous and dynamic. Advertisers may alter their bids at any time. This dataset contains the bids over time of all advertisers participating in Yahoo! Search Marketing auctions for the top 1000 search queries during the period from June 15, 2002, to June 14, 2003. Advertisers' identities and query phrases are anonymized. The data may be used by economists or other researchers to investigate the behavior of bidders in this unique real-time auction format, responsible for roughly two billion dollars in revenue in 2005 and growing [6, 7, 8].

References

[1] Wei Chu, Seung-Taek Park, Todd Beaupre, Nitin Motgi, Amit Phadke, Seinjuti Chakraborty and Joe Zachariah (2009). A case study of behavior-driven conjoint analysis on Yahoo!: Front page today module. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1097–1104.

[2] Lihong Li, Wei Chu, John Langford and Robert E. Schapire (2010). A contextual-bandit approach to personalized news article recommendation. *Proceedings of the 19th International Conference on World Wide Web*, 661–670.

[3] Lihong Li, Wei Chu, John Langford and Xuanhui Wang (2011). Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. *Proceed-*

ings of the Fourth International Conference on Web Search and Web Data Mining, 297–306.

[4] Justin Dyer and Art Owen (2010). Visualizing bivariate long tailed data. Technical report, Stanford University, Statistics.

[5] Abhay Goel and Prerak Trivedi (2012). Finding Similar Music Artists for Recommendation. Under review.

[6] Benjamin Edelman and Michael Ostrovsky (2005). Strategic bidder behavior in sponsored search auctions. In *Workshop on Sponsored Search Auctions*, ACM Electronic Commerce, 2005.

[7] Jia Yuan (2012). Examining the Yahoo! Sponsored Search Auctions: A Regression Discontinuity Design Approach. *International Journal of Economics and Finance*, vol. 4, to appear.

5 Working Groups

On Thursday afternoon, the participants divided into three working groups, each focused upon one of the available datasets. One of those groups subsequently split into two, with each subgroup addressing a different aspect of the data. The working groups are listed below

Working Group 1: Yahoo! Front Page Today Module User Click Log Data.

The working group leader was Zainab Jamal (Hewlett Packard). The members were Douglas Galagae (graduate student, University of Maryland), Giri Gopalan (graduate student, Harvard), Yi Jiang (Rapp, Inc.), Chirag Lakhani (graduate student, North Carolina State University), Broderick Oluyede (professor, Georgia Southern University), and Reginald Roberts (Rapp, Inc.).

Working Group 2: Yahoo! Music User Ratings of Musical Artists.

The working group leader was Tuwana Lamar (Morehouse College). The members were Oleksandr Gromenko (graduate student, Utah State University), Alessandro La Torraca (graduate student, University of Milan), Alan Lenarcic (postdoctoral fellow, University of North Carolina–Chapel Hill), Piaomu Liu (graduate student, University of South Carolina), Vyacheslav Lyubchich (graduate student, University of Waterloo), and Minh Pham (graduate student, Rutgers University).

Working Group 3A: Yahoo! Search Marketing Advertiser Bidding Data.

The working group leader was Xiaoming Huo (professor, Georgia Tech). The members were Tim Hopper (graduate student, North Carolina State University), Zhou Li (graduate student, Notre Dame), Qiyi Lu (graduate student, SUNY Binghamton), and Jacopo Soriano (graduate student, Duke University).

Working Group 3B: Yahoo! Search Marketing Advertiser Bidding Data.

The working group leader was Sarah Tooth (graduate student, Rice University). The members were Hossein Azari (graduate student, Harvard University) and Noah Silverman (graduate student, UCLA).

All four working groups analyzed their data sets and produced draft papers. All four working groups intend to further polish their papers and submit them to refereed journals for publication.

5.1 Working Group 1

The objective was to predict click-through rates on front-page news articles presented to Yahoo! viewers.

Yahoo! has a rotating pool of about 20 articles at a time, and when a user logs in to the Yahoo! homepage, one of those 20 articles is selected for display. Over time, an article becomes stale, is removed from the pool, and replaced by a new article. The dataset has information on the click-through rate, in 5 minute intervals, for each article in the pool over the course of several months. This makes it is possible to track time dynamics.

Additionally, the dataset provides clustering information on both the users and the articles. Specifically, Yahoo! used (private) covariate information on registered users to create five clusters, and the dataset reports the probability of membership for each user in each of the five clusters, via a five-component vector whose entries are nonnegative and sum to 1. We conjecture that these clusters correspond to such things as age, gender, and income, and the probabilities of cluster membership for the non-registered users in the sample are based on recent browsing history. Similarly, the (anonymized) news articles are clustered into five groups, which could represent such categories as international news, celebrities, sports, and so forth, and each article in the sample has a five-component vector showing membership probabilities.

The analysis involved either logistic regression or a Bayesian serving scheme model to predict click-through rates. For logistic regression, the covariates were time of day, probabilities of membership, and interactions among the probabilities of membership. Aggressive variable selection was used, based on the Lasso, to reduce the number of high-order interaction terms. In the Bayesian serving scheme, the estimated total click-through rate was obtained by summing over the click-through rates for each user cluster, weighted by the probabilities of cluster membership. The Bayesian serving scheme had the best behavior, although both methods achieved substantial improvement over baseline article recommendation.

Future work might construct a more thoughtful model for the time dynamics of an article's popularity, and use information on article clusters to ensure that the article pool has sufficient diversity that some article will always be attractive to each user.

5.2 Working Group 2

The objective was to develop a recommender system for musical bands, based on ratings assigned by users.

Most users rated only a small number of bands, and those ratings tended to be extreme—people either rated bands very high, or very low, and many users rated a disliked band as 0 on a scale from 0 to 100.

The analysis strategy was to duplicate the strong regularization (shrinkage) method pioneered by the winning team in the Netflix competition, as applied through content filtering versus collaborative filtering. Two types of collaborative filtering were considered: neighborhood methods and latent factor models.

The data were randomly divided into a training set and a test set. The Netflix strategy, based on the presentation given on Wednesday by Robert Bell, was modified to use an L_1 penalty rather than the original L_2 penalty, to accomodate outliers. (The Netflix ratings ranged from 1 to 5, so outliers had little impact; in this data set, the ratings range from 0 to 100, supporting the use of a robust L_1 penalty.) Other aspects of the Netflix procedure were also modified to suit the circumstances; in particular, the working group attempted to account for heteroscedasticity.

The analysis employed tuning parameters, analogous to controlling sensitivity and specificity through a threshold rule. If the threshold for user satisfaction is too high, there may be no band that can be confidently recommended; if it is too low, then predictive accuracy suffers. The tuning parameters were determined by cross-validation on the training set.

The predictive accuracy on the test set was comparable to that obtained by the Netflix challenge winners.

5.3 Working Group 3A

The objective was to use generalized first-price auction theory to construct a parametric model that describes the distribution of optimal bids, as a function of covariates, such as time of day and key phrase id.

The model was refined based on exploratory data analysis, which incidentally identified bidding patterns that seem to be associated with systematic efforts to probe and learn the market. After some data cleaning, the model was fit using nonlinear non-convex optimization, and the results suggest that the market conforms closely to theoretical predictions for rational behavior in first-price auctions.

The importance of this group's finding is enhanced by the fact that (subject to confirmation) it is reported that the bidding market was a second-price auction. The working group produced an extensive draft paper. From a behavioral economics standpoint, it is interesting that bidders in a second-price auction act as if they were in a first-price auction.

As further work, this group intends to assess fit to the predictive model for a larger number of key phrases. The results from this study interact with those from Working Group 3B, and there was vigorous discussion between these in the closing presentations, as the groups attempted to reconcile their findings.

5.4 Working Group 3B

During exploratory analysis of the Yahoo! Search Marketing Advertiser Bidding Data, this working group noticed cluster structure in the bidding behavior. Further examination showed that this was not a time-of-day effect, and that the clustering was a regular feature.

Their conjecture was that different bidders may have different information or insight on the value of the viewers, and thus make their bids correspondingly.

The group fit a hierarchical mixture model to the data, using a repulsive mixture model to avoid over-reliance upon correct specification of the mathematical form of the mixture components. That model routinely found about three distinct components, and this raises the possibility that the fine structure in bidding strategies reflects important information asymmetries in the market.

This work needs further development, and the closing discussion on this finding suggested a number of possible explanations. The group intends to do further work on this, to confirm their preliminary findings.

6 Feedback and Outcomes

The presentation component of the program was characterized by unusual audience interaction and engagement, full of lively and engaged discussions.

Industry representatives to the program said they were impressed by the quality and capability of the graduate student and postdoctoral participants, and there is the likelihood that this will lead to offers of employment.

Many participants indicated that the program seems to have succeeded in creating a cohort of new researcher statisticians and applied mathematicians who have interest in and understanding of the analytic problems associated with Computational Advertising. The hope is that many of them will continue to work in this area, and develop new methodologies that better serve the needs of this emerging business paradigm. The SAMSI cohort has the potential to contribute substantially to the growth of this economic sector. All four working groups produced extensive draft papers and intend to submit these, after further work, for publication in refereed journals.

Participants in the program have submitted a proposal for an invited session on Computational Advertising to the 2013 Joint Statistical Meetings, sponsored by the American Statistical Association's Section on Statistical Learning and Data Mining.

Nearly all participants are interested in attending the 2014 conference on business analytics that is being jointly organized by the American Statistical Association's Section on Statistical Learning and Data Mining and the International Statistical Institute's International Society for Business and Industrial Statistics. The conference will be held in Durham, NC, and may potentially serve as the front end for a second SAMSI summer program on this general topic area.

7 Appendix: Working Group Writeups

The write ups of Working Groups 1, 2 and 3a are appended.

YAHOO! FRONTPAGE CONTENT RECOMMENDATION

ZAINAB JAMAL, CHIRAG LAKHANI, GIRI GOPALAN, YI JIANG, REGINALD ROBERTS, BRODERICK OLUYEDE, DOUGLAS GALAGATE, NALINI RAVISHANKER

This article is an report of the Yahoo! Front Page Content Recommendation working group during the SAMSI computational advertising workshop.

1. Background

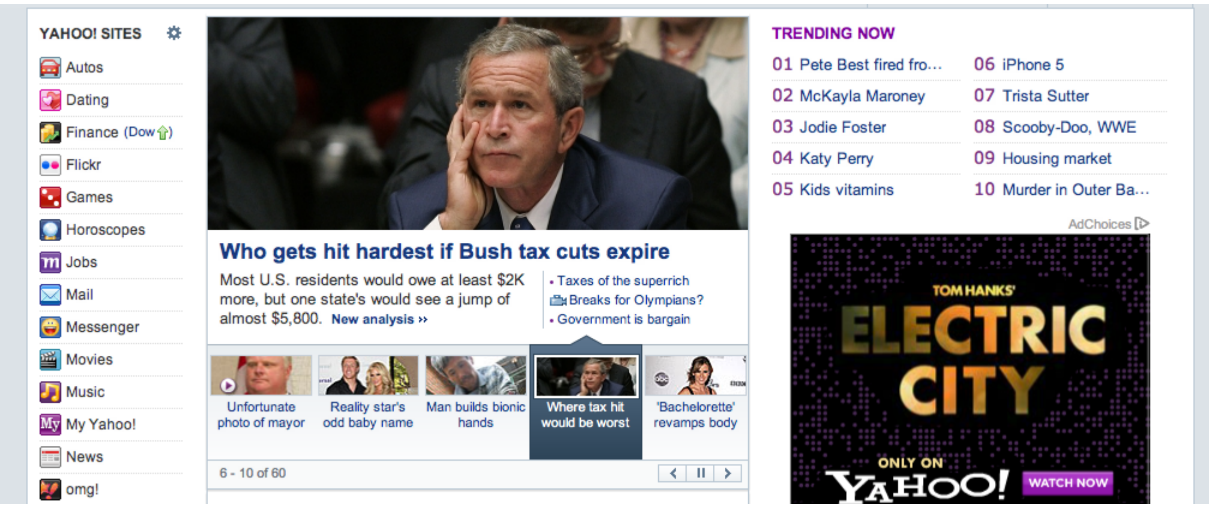
With the proliferation of data a central question for many web-based companies is how does one optimize content on a website to optimize the user's experience. With the explosion of content it is difficult for a user to select the right content. The goal of the content publisher is select the best and most relevant content to attract and retain users on their website. An effective method of retention is to provide personalized content recommendation based on the user's interest. Users generate a great deal of data based on their behavior patterns and demographic information but the problem of delivering optimal content is still difficult. In [Agarwal2] it was shown that implementing a personalized recommendation system on the Yahoo! Front Page news module generated a significant lift in user interaction compared to a simple human editorial based method. A reason cited is that editors select articles that are considered most popular for all users. Such an article may not be of interest to a significant population of web users, therefore it is important to develop a recommendation system that incorporates the users interest as well. While this is a difficult problem the potential payoff for web-based companies is quite significant.

2. Research Problem

We study this problem in the context of the Yahoo! Front Page Today news module. This module is shown in Figure 3. In the problem there are five positions for news articles denoted F1, F2, F3, F4, and F5. The article in position F1 is the most prominent in the module and our goal was to develop a serving scheme for articles so that a user would be induced to click on the story in position F1. The metric of interest is the click through rate (CTR) at a time t for a given article a which is defined by

$$CTR = \frac{\text{total number of clicks at time t for article a}}{\text{total number of views of article a at time t}} \tag{1}$$

2.1. **Dataset.** To give more details about this problem it is important to first discuss the type of data available to study this problem. The datasets is user log data from the Yahoo! Front Page Today Module. There are 10 total datasets each dataset is a snapshot of all user events that occur during one day one Yahoo! Front Page Today Module. The dataset logs, in five minute intervals throughout the day, all of the users on the Yahoo! Front Page at that instance and whether they clicked on the article that was located in position F1. At any given moment there is a pool of twenty articles from which a given user can select from in the front page module. One of these articles is randomly selected and placed in the F1 position. In the dataset, there are approximately 1000 user features that are used to represent user through the analysis these user features it was shown that these can be grouped into five clusters. The user u is represented as a five dimensional vector $u = [u_1, u_2, u_3, u_4, u_5]$ where each u_i represents the probability the user is in the the ith cluster. Similar analysis was done for each article and it was shown that there are five clusters for an



Who gets hit hardest if Bush tax cuts expire

Most U.S. residents would owe at least $2K more, but one state's would see a jump of almost $5,800. New analysis ››

FIGURE 1. Yahoo! Frontpage Module

article and the article is represented by a five dimensional vector $a = [a_1, a_2, a_3, a_4, a_5]$, where a_i represents the probability that article a is in cluster i. In the data file a typical event entry is of the form

1241160900 109513 0 —user 2:0.000012 3:0.000000 4:0.000006 5:0.000023 6:0.999958 1:1.000000 —109498 2:0.306008 3:0.000450 4:0.077048 5:0.230439 6:0.386055 1:1.000000 —109509 2:0.306008 3:0.000450 4:0.077048 5:0.230439 6:0.386055 1:1.000000 [[...more article features omitted...]] —109453 2:0.421669 3:0.000011 4:0.010902 5:0.309585 6:0.257833 1:1.000000

Where the entries correspond to the following:

- 1241160900 - Unix time for this event
- 109513 - article ID for the article served to the user
- 0 - whether the user clicked on this particular article
- user 2:0.000012 3:0.000000 4:0.000006 5:0.000023 6:0.999958 1:1.000000 - user's feature vector (the last entry is always 1)
- 109498 2:0.306008 3:0.000450 4:0.077048 5:0.230439 6:0.386055 1:1.000000 - feature vector for article 109498 (one of the articles in the pool of articles)

The entries after the user vector are the article feature vectors for all of the articles in the available content pool for this particular user.

2.2. **Problem Statement.** Given such data the problem for this working group is to develop a serving scheme in order to serve a user with feature vector u the best article a out of a pool of twenty available articles in order to maximize CTR. The solutions we investigate must, of course, take into account both article and user information given by their corresponding feature vectors. It must also incorporate time of day affects. As will be shown in the exploratory analysis user behavior varies based on time as well as user interests.

3. Exploratory Analysis

3.1. **User/Article Clusters.** As stated in the previous section both user and article features were grouped into 5 clusters. The five dimensional user vector u was derived from 1000 categorical components starting from demographic information i.e. gender, age, geographic location, and behavior information. These features were aggregated and found to be in five clusters. The feature vector u

represents the probability of belonging to each cluster. The article vector a has feature dector such as article categories inferred from the source and editor. They are also clustered into 5 clusters.

3.2. **Data Analysis.** Initially, exploratory data analysis was done in order to understand click rate dynamics.

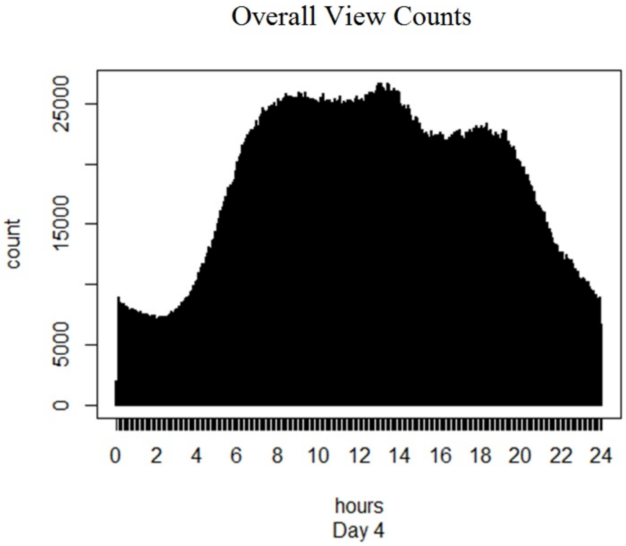


FIGURE 2. Aggregate CTR for One Day

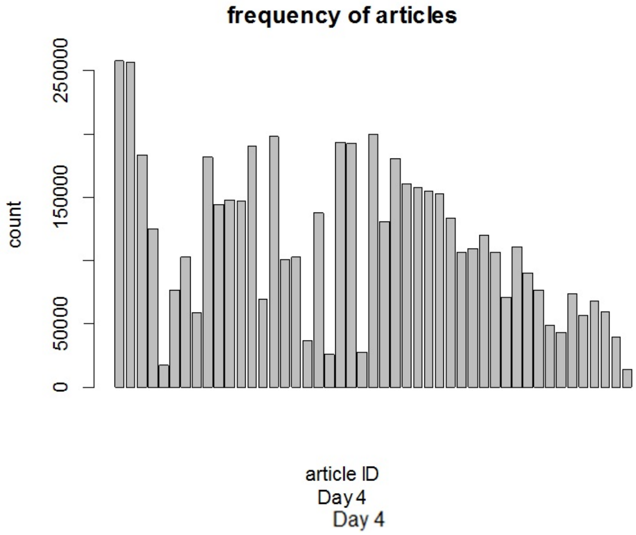


FIGURE 3. Frequency Article was Selected

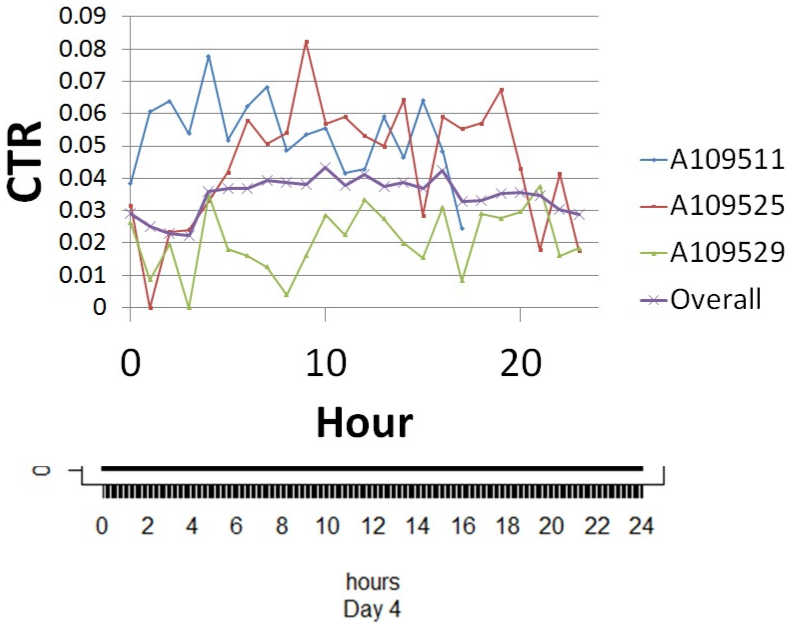


Figure 4. CTR by Article

4. Methods

4.1. **Logistic Regression.** A critical component in any article serving scheme is to estimate the probability a user will click on a given article. One such method for developing a probability distribution is to use logistic regression. In logistic regression we model the probability p that a user $u = [u_1, u_2, u_3, u_4, u_5]$ that is served an article $a = [a_1, a_2, a_3, a_4, a_5]$ at time t will click on the link i.e p is the CTR. Logistic regression is considered a generalized linear model where given inputs $x_1, x_2, x_3, \ldots, x_n$ one can do regression for categorical variables. More precisely, the log odds of p are given by

$$\log \frac{p}{1-p} = \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n \tag{2}$$

The inputs for this model consist of time t, user features u, and article features a. Initially we experimented with the inputs being linear in u and a but having both a linear and quadratic term in t. The results of this experiment are given in section 5. This model was extended by including nonlinear terms of the form user-article, user-time, article-time, and time-article-user. In section 5 we will explain the methodology for sampling in order to train the logistic regression for prediction.

Once the logistic model has been trained on the data, the article, the serving scheme consists of choosing the article that has the highest CTR rate p. More precisely at time t a user u has to select an article from a collection of articles $a \in \mathcal{A}$. For each article $a \in \mathcal{A}$ the algorithm inputs a, u, and t into the logistic regression model and selects the article a that gives the highest probability. It can be seen that this method attempts to incorporate user features, article features, and time of day effects into the model. This process can be summarized as follows:

(1) Acquire sample data to train logistic regression model
(2) Select the features and/or combination of features from time t, article features a, and user features u that will be used in model.
(3) Train logistic regression model on the data.

(4) Given a new user u at time t, calculate the probability p of clicking for each article a in the available pool of articles $\mathcal{A}$.

(5) Pick the article a that has the highest probability p for user u.

4.2. **Bayesian Story Serving Scheme.** We have developed a novel methodology which makes direct use of the unique feature space existent in the Yahoo! Front Page dataset. Precisely, associated with each user that visits the site is a five dimensional vector $[p(C_1), p(C_2, p(C_3), p(C_4), p(C_5)]$ where each component of the vector is the probability of the user belonging to a cluster C_i, or $p(C_i)$ for short. Assume that the content pool consists of m articles $\in \{A_1, ...A_m\}$. Furthermore, assume at a time t, we may estimate the conditional probability that a user in cluster i will click on article j, $p(A_j|C_i)$. Since the clusters $C_1, C_2, ...C_5$ (theoretically) partition the space of users, the law of total probability gives the probability of a user clicking on a particular add, $P(A_j) = \sum_i P(A_j|C_i)P(C_j)$. To be explicit, $\sum_i P(A_j|C_i)P(C_j) = \sum_i \frac{P(A_j,C_i)}{P(C_i)} P(C_j) = \sum_i P(A_j, C_i) = P(\cup_i(A_j \cap C_i)) = P(A_j \cap \cup_i C_i) = P(A_j)$.

We can succintly restate this relationship as follows. Let A_t be an m by n matrix for which entry a_{ij} contains an estimate of $P(A_i|C_j)$ at time t, and let $u \in \mathbb{R}^{\ltimes}$ be a user vector we are trying to predict which add to serve to. The "Bayesian Serving Scheme" serves the story corresponding to the maximal component of $A_t u \in \mathbb{R}^{\gtrdot}$.

To estimate the matrix A_t, we look at all events in the the last 5 time steps, sample cluster members which we define as users with a feature greater than .9, and determine the likelihood of clicking of the cluster representatives on all stories.

5. Results

5.1. **Training and Simulation.** To test our serving schemes, we trained a logit model and the Bayesian Serving Scheme model using 45 minutes worth of Yahoo! front page log data, from Unix Time Stamp 1241166600 to 1241169000. Evaluating our serving schemes presented a challenge for the simple reason we could not actually simulate them in real time on the Yahoo! front page. To circumvent this, we examined the events during time stamp 1241169300, and subselected a set of events for which the story served was the one predicted by our schemes. By computing the CTR for these subsets (precisely one for each of the schemes) of events, we were able to procure a reasonable estimate of the CTR lift provided by either scheme, in comparison to the overall CTR during the time period.

5.2. **Results.**

Serving Scheme	**CTR**
Random	.02565418
Logit	.04093567
BSS	**.04710144**

Table1: CTR Comparison of Serving Schemes

5.3. **Discussion.** Our preliminary validation seems to show that both the logit and BSS schemes severely outperform the random serving scheme, while the BSS outperforms the logit serving scheme. At the outset, a hypothesis for this behavior is that the BSS explicitly builds the structure of the user feature space into its formulation, whereas the logit model treats the features as any generic set of vectors. Of course, more rigorous tests, spanning significantly more time points are necessary to confirm this hypothesis.

6. Future Work

We feel this work can be extended in many different suggestions. It was suggested, at the beginning of the workshop, that an effective method of modeling this problem maybe using a Bayesian dynamic generalized linear model. A members of our group (Nalini Ravishanker) has R-INLA code that can be modified in this context. Throughout this week we have developed a method of handling and modeling this data and it is now possible to find model this problem using more advanced models such as Bayesian DGLM. A difficulty in modeling the time dynamics for this problem comes from the life cycle of an article. While time is obviously an important feature, there may be other dynamical features that can be incorporated in order to accurately capture the time dynamics. Incorporating variables such as article lifetime or time to max may be quite helpful. More exploratory work needs to be done regarding features.

References

[Agarwal1] D. Agarwal, B. Chen, P. Elango *Spatio-Temporal Models for Estimating Click-through Rate,*, The 18th International World Wide Web Conference. (2009).

[Dunson1] D. Dunson, A. Herring *Semiparametric Bayesian Latent Trajectory models*, NIH Report. (2012).

[Agarwal2] D. Agarwal, B. Chen, P. Elango, R. Ramakrishan, N. Motgi, S. Roy, J. Zachariah *Online Models for Content Optimizations*, NIPS. (2008).

[Wang1] X. Wang, W. Li, Y. Cui, R. Zhang, J. Mao *Click-Through Rate Estimation for Rare Events in Online Advertising*, Yahoo! Technical Report. (2009).

[Li1] L. L, W. Chu, J. Langford, R. Schapire *A Contextual-Bandit Approach to Personalized News Article Recommendation*, arXiv:1003.0146 [cs.LG]. (2012).

Application of Penalized Matrix Factorization to Yahoo! Music Data

Oleksandr Gromenko, Tuwaner Lamar, Alessandro La Torraca,
Piaomu Liu, Vyacheslav Lyubchich and Minh Pham

August 17, 2012

Abstract

E-commerce has flourished over the last few years, huge volumes of web data are generated and are to be analyzed. Recommender systems aim to match best products with the customers. Quality of recommendations plays a significant role in market place. Matrix factorization models are a type of latent factor models that has shown its strength in coping with the problem of matching consumers with the best products. However, recommendations become difficult when available data are massive and sparse. Such datasets include Yahoo! Music data. The goal of this project is to create a statistical procedure for matching Yahoo! Music users with artists based on their past rating behaviour. Performance of the statistical procedure is evaluated using numerical simulations.

1 Introduction

Recommender systems are one of the most important components for internet based merchants, entertainment websites, social networks and many others. There are several types of underlying statistical approaches for recommender systems such as latent factor model (LFM) and k-Nearest Neighbor (kNN) algorithm. kNN is a strategy that has been used by researchers and e-commerce recommendation systems. The fundamental assumption of kNN is that if two users have the same ratings on a certain number of items they tend to rank similarly other items. It works in the same manner for any two items.

LFM is an approach that categorizes users and items based on a certain number of factors inferred from user-item rating patterns. For instance, a movie such as *The Terminator* will score high on the action and science-fiction factors and low on romance and comedy. Matrix factorization is one of the most successful realizations of LFM. For example, the winners of the Netflix Prize competition applied the matrix factorization models to rating data of movie viewers, and demonstrated its superiority to some classic approach to provide product recommendations Koren *et al.* (2009).

Usually, commercial datasets are matrices with columns representing certain items such as movies, artists, etc. and rows representing users. Entries of a matrix are ratings given by a user to a specific item. Commercial datasets, such as Netflix movie dataset and Yahoo! Music dataset, contain a very large number of users and items, but relatively small number of ratings. Sparsity of data is one of the main challenges towards building an efficient recommender system.

The goal of this project is to make a recommender system for Yahoo! Music dataset similar to Cinimatch (the recommender system for Netflix). Essentially, a recommender system is a two step statistical procedure. The first step is to impute missing observations and the second step is to decide whether to recommend a certain item or not.

The rest of the paper is organized as follows. In section 2 we provide detailed description of Yahoo! Music dataset and explain reduction of the data. Next in section 3 we discuss a statistical model for a recommendation system. Results of estimation as well as simulation study is reported in section 4. Finally, in section 5 we discuss possible future work which can lead to publication.

2 Data

In this section we provide a detailed description of the data, obtained from the Yahoo! Webscope library (http://webscope.sandbox.yahoo.com). We use the data set R1 – Yahoo! Music User Ratings of Musical Artists, version 1.0. It is a snapshot of the Yahoo! Music community's preferences for various musical artists, collected during one month period sometime prior to March 2004. The data are represented by a matrix, containing users ids, codes of the artists and corresponding ratings given by users to particular artists. The actual names of the musical artists matching to the codes are stored in a separate file. These data sets have been reviewed to conform to Yahoo!'s data protection standards, including strict controls on privacy, and may be used by researchers to validate recommender systems or collaborative filtering algorithms.

The size of original data set is 423 MB. It contains over eleven million ratings, ranging from 0 to 100, made by almost two million users for a list of 98,211 musical artists. The distribution of artists by the number of ratings (Figure 1) reveals that the majority of artists got very few ratings. The average number of votes in the entire dataset is 59 per user. While the most active user of the dataset has rated 97993 artists. The corresponding user-by-artist rating matrix is about 99.9% sparse. An extensive amount of time and (or) computational power is required to perform the analysis. Severe sparsity of the dataset is comparable to that of the Netflix data. However, the Yahoo! Music data present a greater challenge for analysis since possible bias is larger when the range of the rating scale is wider.

To overcome the problem of sparsity and reduce dimensionality, the data were preprocessed in the following way:

1. Records of about 40,000 first users in the data set were analysed in order to determine some 1,000 users with the largest amount of given ratings, i.e. the most active users, (see Algorithm 1 for details).
2. Based on the ratings of 40,000 first users, 1,000 the most frequently rated musical artists were chosen (Algorithm 2).
3. The subset for analysis was defined as a matrix 1,015×1,005, containing ratings of the most frequently rated musical artists by the most active users.

In addition, the rating value of 255, corresponding to users' choice "Never play again", were replaced by zeroes to avoid categorical observations. Note, we did not analyze the entire data set because time constrain.

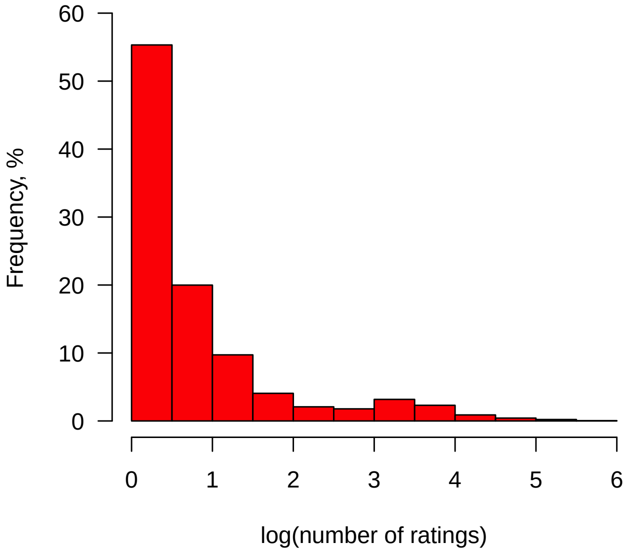


Figure 1: Frequency histogram for artists in the data set R1, by number of given ratings (in a log-scale).

Algorithm 1: Users counter pseudocode

```
Data: ydata-ymusic-user-artist-ratings-v1_0.txt
Result: Produce the list of most active users
open TextSource for input;
int [] users = new int[users number];
while readLine(TextSource) != null & UserID < 40001 do
    users[ readLine[userID]]++;
end
```

Algorithm 2: Popular artists pseudocode

```
Data: ydata-ymusic-user-artist-ratings-v1_0.txt
      Most Active Users List
Result: Produce the list of most ranked artists
open TextSource for input;
int [ ] votes = new int[artists number];
HashMap map = new HashMap;
while readLine(TextSource) != null & UserID < 40001 do
    if map.contains(readLine[artist]) & UserID is in Most Active Users List then
        Votes[map.get(readLine[artist])]++;
    else
        votes[map.size+1]++;
        map.push(readLine[artist], map.size+1);
    end
end
```

The Algorithms 1 and 2 allowed us to obtain a relatively small matrix with sparseness reduced to 87.0%. Figures 2 through 5 provide some visualizations of the selected subset. The histogram in the Figure 2 reflects the fact, that many ratings, given by Yahoo! Music users, are close to 30, 50, 70 and, especially, to 0 and 100. The rise in the Figure 3 indicates, for instance, that 200 of the most active users account about 44.3% of given ratings in the selected subset. Similarly, 200 of the most frequently rated music artists attract about 52.3% of users' ratings. We should note, that music artist can be rated without having been listened to. The most rated musical artist was rated by 785 users, while the least rated artists were rated only once.

Figure 5 provides histograms for the artist mean ratings and the user mean ratings. The reason for the evident mismatch between these two histograms is due to users personal bias. For example, 3.1% of users gave ratings with the mean value from 0 to 9 (inclusive).

In general, the selected data set contains significant variation. While the average number of ratings per user is 131, and the average number of ratings per musical artist is 132, two users rated more than 900 artists, and another two – less than 10 artists. Similarly 7 artists were rated 600 times or more, and 5 artists – less than 10 times. The consequence of this variation implies the issues in achieving high predicting accuracy of ratings, although it is a common problem for extensive collaborative filtering.

3 Statistical methodology

Throughout the paper we use the following notations: r_{ij} denotes particular rating given by user i for a artist j; I and J represent the total number of users and the total number of artists, respectively, in the selected subset. The goal of this project is to make a music

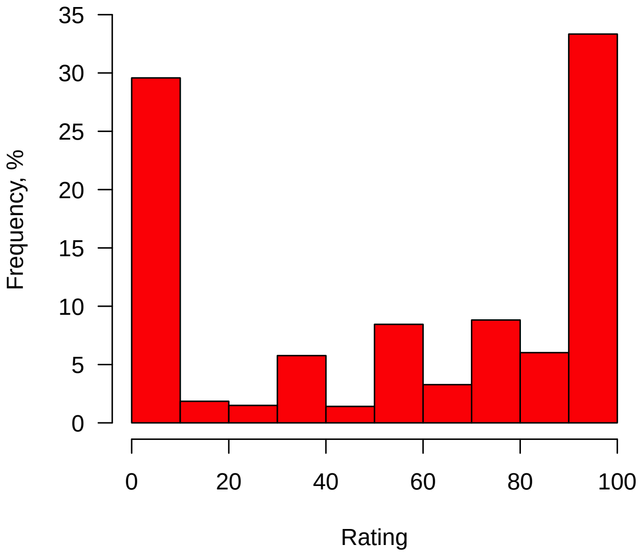


Figure 2: Frequency histogram for ratings in the selected subset. The histogram cells are left-closed intervals: $[0, 10)$; $[10, 20)$; $\ldots$; $[90, 100]$.

recommender system similar to the one proposed for Netflix data, Koren *et al.* (2009). Essentially, a recommender system is a two step statistical procedure. The first step is to impute missing observations and the second step is to decide whether to recommend a certain item or not.

As proposed by Koren *et al.* (2009) each ranking can be represented in the following simple way:

$$\hat{r}_{ij} = \mu + a_i + b_j + \sigma_{ij}, \tag{1}$$

where μ is the global average, a_i is user bias, b_j is artist bias, and σ_{ij} is user-artist interaction.

When the data matrix does not contain missing observations, estimation of the components of (1) is a standard procedure. First, estimate the global mean as an average of all elements, and subtract it from data. Next, estimate user and artist biases as a column and row averages. And the last step is estimation of the interaction.

To proceed let us introduce $I \times J$ matrix $\mathbf{\Sigma}$ of rank K consisting of elements σ_{ij}. Matrix $\mathbf{\Sigma}$ can be represented using Singular Value Decomposition (SVD):

$$\mathbf{\Sigma} = \mathbf{UDV}^T, \;\; \mathbf{UU} = \mathbf{I}_I, \;\; \mathbf{VV} = \mathbf{I}_J, \;\; d_1 \geq d_2 \geq \ldots \geq d_K \geq 0. \tag{2}$$

Let $\mathbf{u}_k$ denote column k of $\mathbf{U}$, $\mathbf{v}_k$ denote column k of $\mathbf{V}$, and note that d_k denotes the kth diagonal element of the diagonal matrix $\mathbf{D}$. Then, using Eckart–Young Theorem for any

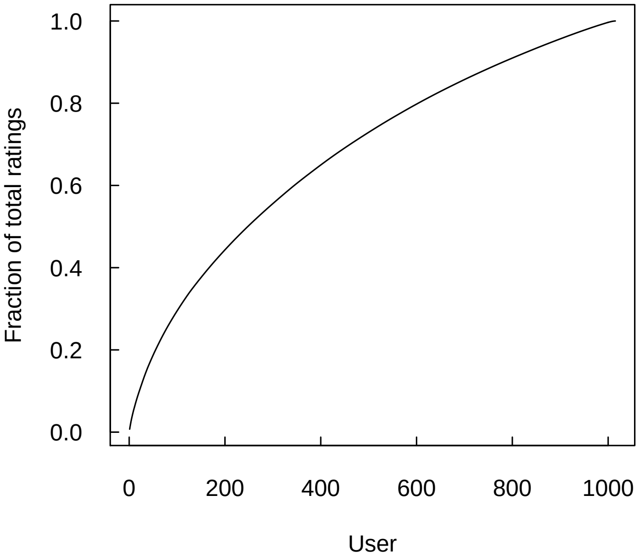


Figure 3: Cumulative proportion of ratings in the selected subset, by users. Users are on the horizontal axis, sorted by the number of artists rated (from most to least).

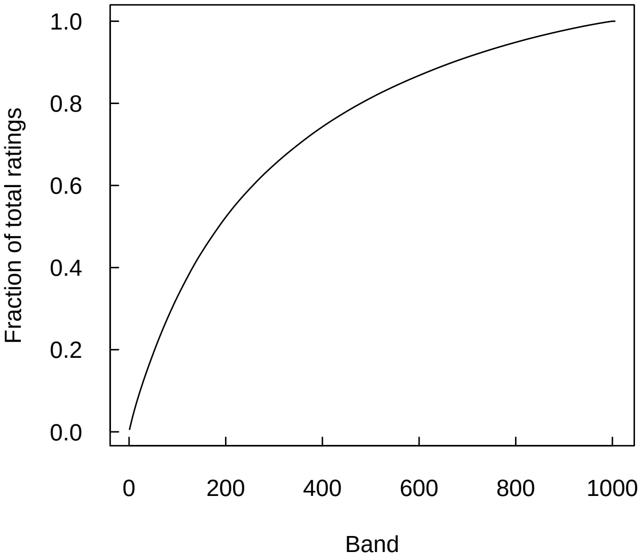


Figure 4: Cumulative proportion of ratings, by artists, for the selected set. Artists are on the horizontal axis, sorted from most to least rated.

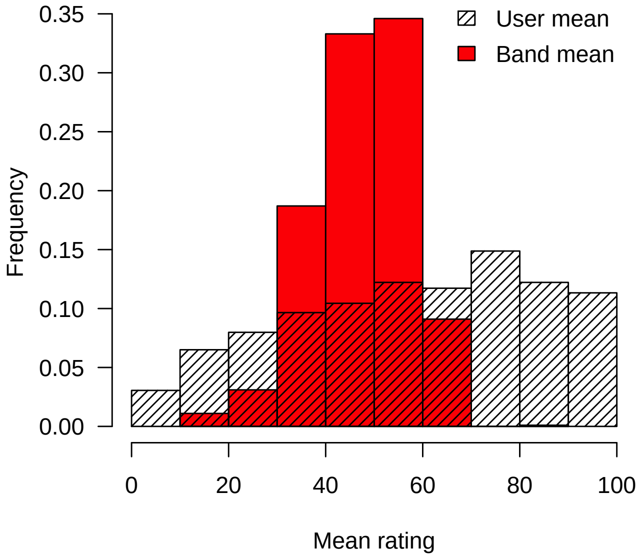


Figure 5: Histograms for mean artist ratings and mean user ratings in the selected subset. The histogram cells are left-closed intervals: $[0, 10)$; $[10, 20)$; $\ldots$; $[90, 100]$.

$r \leq K$,

$$\sum_{k=1}^{r} \hat{d}_k \hat{\mathbf{u}}_k \hat{\mathbf{v}}_k^T = \arg \min_{\hat{\mathbf{\Sigma}} \in M(r)} \|\mathbf{\Sigma} - \hat{\mathbf{\Sigma}}\|_F^2, \tag{3}$$

where $M(r)$ is the set of rank-r $I \times J$ matrices and $\|\cdot\|_F^2$ indicates the squared Frobenius norm (the sum of squared elements of the matrix). In other words, the first r components of the SVD give the best rank-r approximation to a matrix, in the sense of the Frobenius norm. Note that a particular element of $\hat{\mathbf{\Sigma}}$ is equal to

$$\hat{\sigma}_{ij} = \sum_{k=1}^{r} d_k \mathbf{u}_k(i) \mathbf{v}_k^T(j).$$

See Feuerverger *et al.* (2012) and Witten *et al.* (2009) for a broader discussion.

When the data matrix does not contain missing observations the above procedure is well-defined and implemented in many software packages. Meanwhile when the data matrix contains large amount of missing observations (sparse matrix) SVD is not well-defined. Also, averaging over all elements as well as over rows and columns may lead to poor estimates of the grand mean and biases. A possible remedy is to impose certain constraints on the unknown parameters. Specifically, the unknown parameters should be estimated by minimizing the following function:

$$\sum_{ij} \{r_{ij} - \mu - a_i - b_j - d\mathbf{u}(i)\mathbf{v}^T(j)\}^2, \text{ subject to } P(\mathbf{v}) \leq c_1,\ P(\mathbf{u}) \leq c_2,\ d \geq 0, \ldots \tag{4}$$

Here $P(\cdot)$ represents some convex penalty function, such as L_p norm ($L_p(\mathbf{u}) = (\sum_i |u_i|^p)^{\frac{1}{p}}$).

Generally, minimization of (4) is difficult and a time consuming procedure. Thus, to be able to finish our project in the time allotted we took a simplified path. We estimate unknown parameters μ, a_i, b_j as averages, which is acceptable approximation for the selected dataset. Next we find parameters d, $\mathbf{u}$, $\mathbf{v}$ by minimizing

$$\sum_{ij}\{r_{ij} - d\mathbf{u}(i)\mathbf{v}^T(j)\}^2, \text{ subject to } \|\mathbf{v}\|_2^2 \leq 1,\ \|\mathbf{u}\|_2^2 \leq 1,\ \|\mathbf{v}\|_1 \leq c_1,\ \|\mathbf{u}\|_1 \leq c_2\,. \qquad (5)$$

Free parameters c_1 and c_2 are found by cross-validation. For estimation we use PMD(L_1, L_1) function implemented in R package PMA, Witten *et al.* (2009).

4 Results

To validate our recommender system we split the dataset into training and testing subsets in the following way. We randomly select $IJ/2$ elements and put them into a training matrix (at the same locations) and other elements we put into a test matrix (at the same locations). Missing values in both matrices are filled with NAs (NA is a standard notation for a missing element in R). In such a way we obtain two $I \times J$ matrices. Note that the above procedure may not work for a very sparse matrix. Next, we estimate the unknown parameters of (1) using training matrix. Then we calculate RMSE for both training and test matrices:

$$\text{RMSE} = \frac{1}{M}\sum_{ij}(r_{ij} - \hat{r}_{ij})^2,$$

where M is the number of available observations (either for training or testing matrices). Note that RMSE is slightly different every run since the assigning process is random. Typically, RMSE for a training set is 775.03 and 647.85 for a testing set.

Now, we need to decide whether to recommend a particular artist to a certain user or not. Intuitive approach is to introduce some cutoff value. If a predicted rating exceeds this cutoff value system will automatically recommend an artist. It's clear (illustrative simulations are provided below) that a cutoff is basically an interplay between the amount of recommendations and user satisfaction. A large cutoff leads to a high satisfaction rate and small number of recommendations. Since the number of recommendations is directly related to the number of listenings/purchases it might negatively affect revenue. On the other hand, a small cutoff leads to a large number of recommendations and large unsatisfactory rate. Large unsatisfactory rate might, also, negatively affect revenue since some users could decide not to use a particular website anymore. See below Fig. 6, Fig. 7 and Fig. 8.

Now, we explain details of a simulation study. For simulation study we further minimize the dataset by selecting 100 the most active users and 100 the most rated artists among these users using the algorithm above. Again we randomly split data matrix on learning and testing subsets. Then we predict rankings for testing subset and compare them with the actual values. We are interested in three quantities: satisfaction rate, proportion of "unhappy users" and the number of recommendations per user. Satisfaction rate is defined as a proportion of correct matches among all users. Proportion of "unhappy users" is a ratio of a number of recommendations which users do not like and total number of recommendations. And the recommendation rate is number of all recommendations per user. Results of simulational study are shown in Fig. 6, Fig. 7 and Fig. 8.

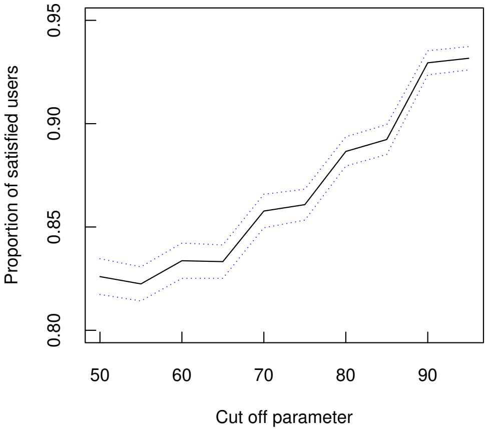


Figure 6: The satisfaction rate for users as a function of the cut off parameter. Blue dotted lines represent 95% confidence interval.

5 Future work

There are several possible extensions which could be done in the future. The first thing is to compare performance of our imputation technique and the Netflix winning competition method, Koren *et al.* (2009). Explore selection of the cutoff value and its influence on revenue. Finally, we need to implement the imputation procedure suitable for large datasets, possibly using parallel computing and analize the whole Yahoo! Music dataset.

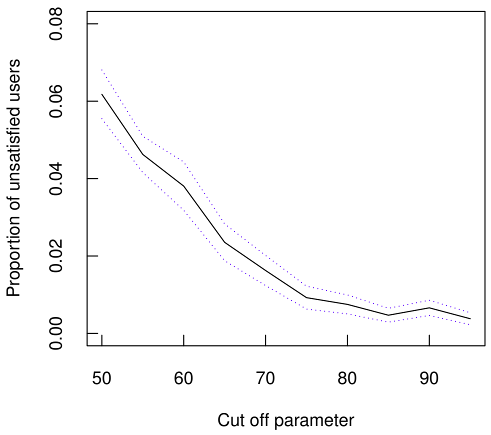


Figure 7: The proportion of “unhapy” users as a function of the cut off parameter. Blue dotted lines represent 95% confidence interval.

References

Feuerverger, A., He, Y. and Khatri, S. (2012). Statistical significance of the netflix challenge. *Statistical Science*, **27,** 202–231.

Koren, Y., Bell, R. and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *IEEE Computer Society*, **42,** 30 − 37.

Witten, D. M., Tibshirani, R. and Hastie, T. (2009). A penalized matrix decomposition with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, **10,** 515–534.

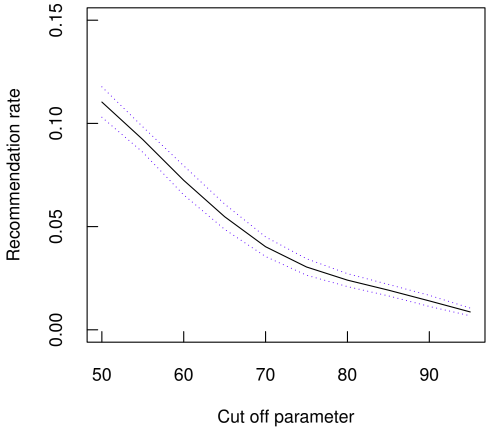


Figure 8: Recommendation rate (number of recommendations per user) as a function of the cut off parameter. Blue dotted lines represent 95% confidence interval.

Can We Use an Equilibrium Model to Interpret the Auction Data from Yahoo! Webscope?

SAMSI Working Group on the Bidding Data Set

August 16, 2012

Internet advertisement and the corresponding auction design is an upcoming field with many interesting problems and huge revenues (Edelman et al., 2007; Edelman and Schwarz, 2010). In this working paper, we study Version 1.0 of Yahoo! Search Marketing advertising bidding data, provided as part of the Yahoo! Research Alliance Webscope program. The data set contains all bids placed on the top $1,000$ keywords during the time period June 15, 2002 to June 14, 2003. There are 10475 bidders in total. The keyword and bidders are coded, so that their true contents and identities are hidden. Bid data are sampled in 15 minute time increments. Some exploratory data analysis (EDA) discoveries are presented in Section 1.2.

The goal of this study is to try to come up with quantitative models for the bidders' behavior, assuming that the market is efficient, so that every bidder eventually is bidding in the optimal way (the one in which they maximize their profit).

We come up with the following model to start with. We assume that there is one auction per keyword. For a prescribed keyword, let p_{tj} denotes the bidding price of the jth bidder at time t. This is the information that is provided in the aforementioned Webscope data set. Based on the above information, we can infer k_{tj}, which is the position that the jth bidder occupies at time t as the result of the position auction.

We assume that for the entire data set, the Generalized First Price (GFP) model is adapted. Though, in the literature, some references, e.g., Yuan (2012), seem to indicate that the GFP was switched to the generalized second price model (GSP) on June 26, 2002 in this data set, however opposite description can be found in Edelman and Ostrovsky (2007). In our data analysis, we decide to believe that the auction scheme is GFP throughout the entire time period.

If every bidder is trying to maximize their profits in this auction. Suppose that they have done so, at the same time, an equilibrium has been achieved. Some game theoretical analysis will end up with the following equations, which should be hold for all time t and bidder j:

$$p_{tj} = q_j - \frac{G_{k,j}}{G_{k-1,j}}(q_j - b^{k+1}), \tag{1}$$

where q_j is the expected return the bidder j suppose to get per click, $G_{k,j}$ and $G_{k-1,j}$ are the number of clicks that the jth bidder will get if she occupies the kth or the $(k-1)$st position as a result of this auction, $k = k_{tj}$ is the position that bidder j occupies at time t (note that k is determined by t and j). Information regarding the game theory that leads to (1) will be articulated in Section 2. Notation $b^{k_{tj}+1}$ is the bidding price by the bidder who takes the immediately lower position. Note that p_{tj} and k_{tj} are available or obtainable from the Webscope data set. Consequently, we can find out $b^{k_{tj}+1}$, which is the bidding amount by the bidder who is immediately below in position. If we introduce another notation: $\lambda_{kj} = G_{kj}/G_{k-1,j}$, then (1) can be rewritten as follows: $\forall t, j$,

$$p_{tj} = q_j - \lambda_{kj}(q_j - b^{k_{tj}+1}). \tag{2}$$

Note that q_j and λ_{kj} are unknowns.

We have observed that the bidders' behaviors are quite heterogeneous. In particular, nobody seem to audit using some sophisticated game theory approach. Instead, there are a lot of rudimentary strategies that are adopted by these bidders. More details on this regard will be reported in Section 1.2. So we can only hope that the necessary condition for equilibrium, which is depicted by (2) is only achieved in an approximate sense. To quantity this idea, we came up with the following optimization framework:

$$\begin{aligned} \min_{\varepsilon_{tj}, q_j, \lambda_{kj}} \quad & \sum_{t,j} \varepsilon_{tj}^2, \\ \text{subject to:} \quad & p_{tj} + \varepsilon_{tj} = q_j - \lambda_{kj}(q_j - b^{k_{tj}+1}), \\ & 0 < q_j, \text{ and } 0 < \lambda_{kj} \leq 1; \end{aligned} \tag{3}$$

that is, after adding some hopefully small perturbations ε_{tj} on the observed data, we can achieve the equilibrium. Note that without the term ε_{tj}, the condition in the above optimization problem is exactly the equation (2). A similar idea is called perturbation in Auerbach et al. (2008).

We will discuss the numerical strategies to solve for (3) in Section 3.

1 Exploratory Data Analysis

Bid data are sampled in 15 minutes time increments. Bids from the same bidder on the same keyword within a 15 minutes interval are indistinguishable. Thus, for each keyword, we assume each 15 minutes interval contains a single auction where each bidder bids the average price of all his actual bids in that interval.

Because of the large number of keywords, we focus on the most popular one and provide some exploratory data analysis for it. Similar behaviours are observed for the other keywords. First we analyze the bid behavior on a year scale, later we focus on the bids pattern within a day.

1.1 Year Scale

On the keyword we consider there are 286 different bidders. Two thirds of the bidders bidded less 10 times in the year, and only 3% of the bidders bidded more than 1000 times. Bidding more than 1000 time means that this bidder bids roughly 3 times per day. In a year there are 35040 auctions.

less than 10 times	66%
between 10 and 100 times	22%
between 100 and 1000 times	9%
more than 1000 times	3%

In Figure 1 we provide a scatterplot of the average and maximum number (over a day) of unique bidders on an auction. The number of bidders is overall constant and relatively small throughout the year. We can identify a daily pattern (see Figure 2); as we might expect, the biggest auctions are from the morning to midnight. The bids price doesn't follow the same pattern of the number of bidders (see Figure 3), but they are constant over the day. We can argue that the prices are constant because the system has reached a stationary equilibrium.

There is no evidence in the data of weekly pattern in term of bid price and number of bidders per auction.

The prices in a auction are close to each other. In Figure 4 we provide the ratio between the 2nd and the 1st price and the 3rd and 2nd price, respectively. Only 11% and 13% of the time the ratio is smaller than 0.9.

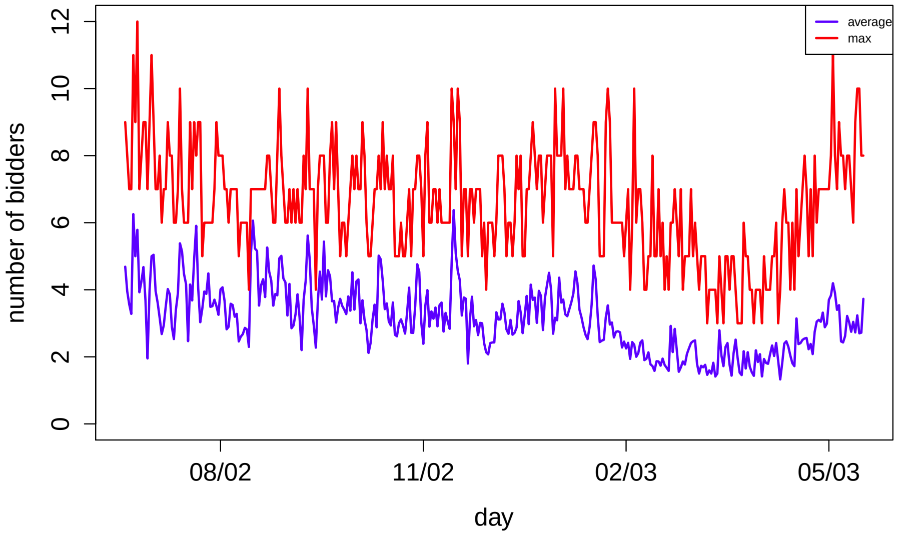


Figure 1: Scatterplot of the average (blue line) and maximum (red line) number of unique bidders on an auction over a day.

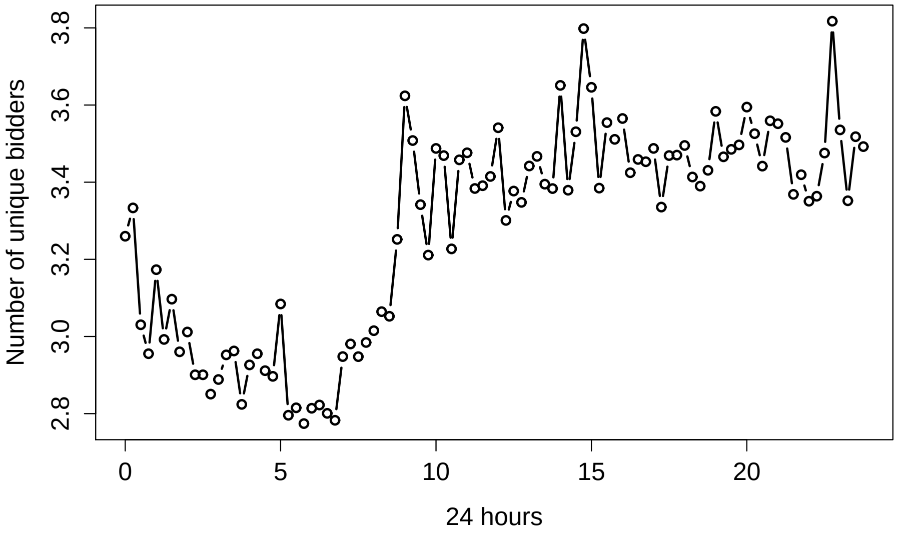


Figure 2: Scatterplot of the average (over a year) of the number of bidder per auction. The highest number of bidders is between 9am and midnight.

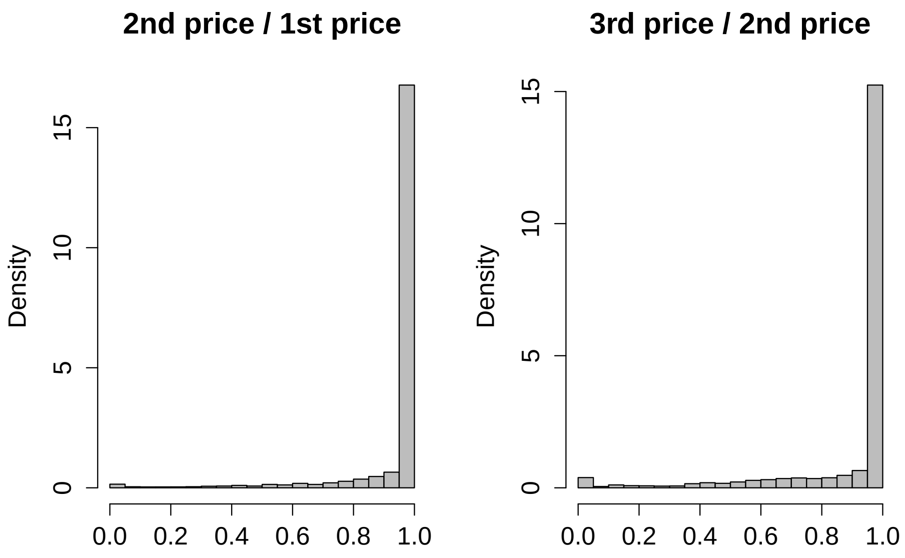


Figure 3: Histograms of the ratio between the 1st - 2nd price and 2nd - 3rd price, respectively. The 3 highest prices are generally very close to each other.

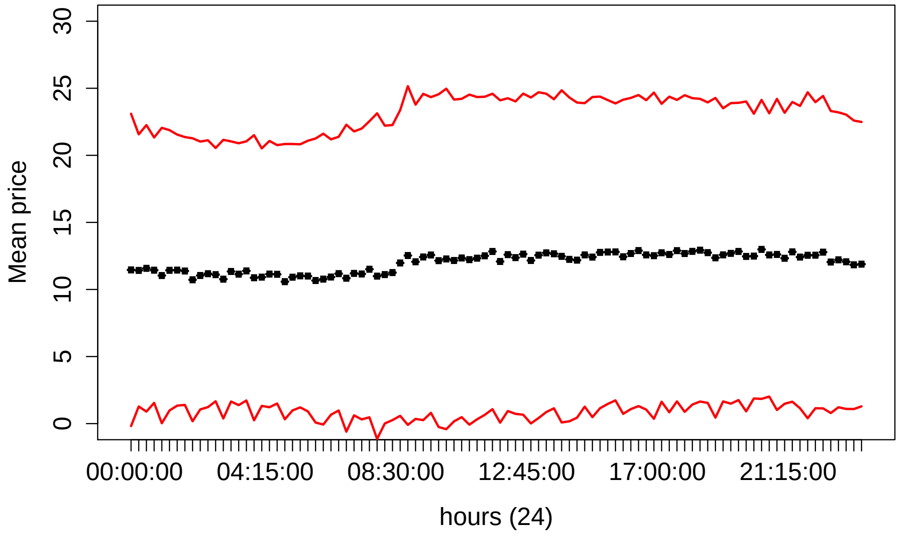


Figure 4: Scatterplot of the mean bidding prices and two standard deviations below and above. (Mean over all days and all bids in the corresponding 15 mins interval.) It shows little variation.

1.2 Year Scale (POLISH)

Many bidders used a cyclic pattern to try to gain from the auction scheme. Edelman and Ostrovsky (2007) studied this kind of bidder behavior. The Figure 5 shows the bidding prices of ten most active bidders on November 15, 2002 for the keyword coded 1.

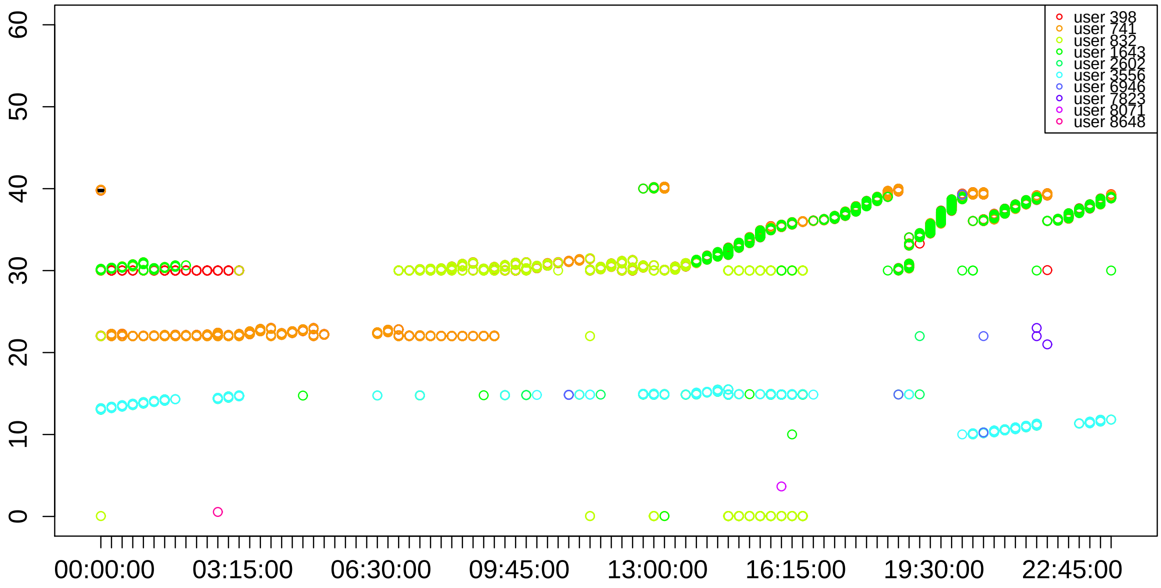


Figure 5: Ten bidders and their bidding prices over one day. The cyclic behavior by several bidders are clearly visiable. It is also observed that the bidding is sparse for most of the bidders – they bade in a small proportion of these auctions. There is one auction every 15 minutes. The total number of auctions is 96.

The Figure 6 display the relation between the bidding prices and positions for the same day and keyword. It seems that there is no evident relation between the bidding prices and position. This may indicate that the position is largely determined by the competition (i.e., how the others are bidding at that time).

We arbitrarily chosen a day: November 15, 2002. There are 701 phrases that are auctioned on this day. We handpicked the phrase that is coded 1. There are 14 bidders who have bidden on this phrase. Figure 7 displays the bidding patterns of these 14 bidders.

Figure 8 shows the highest three prices over the time.

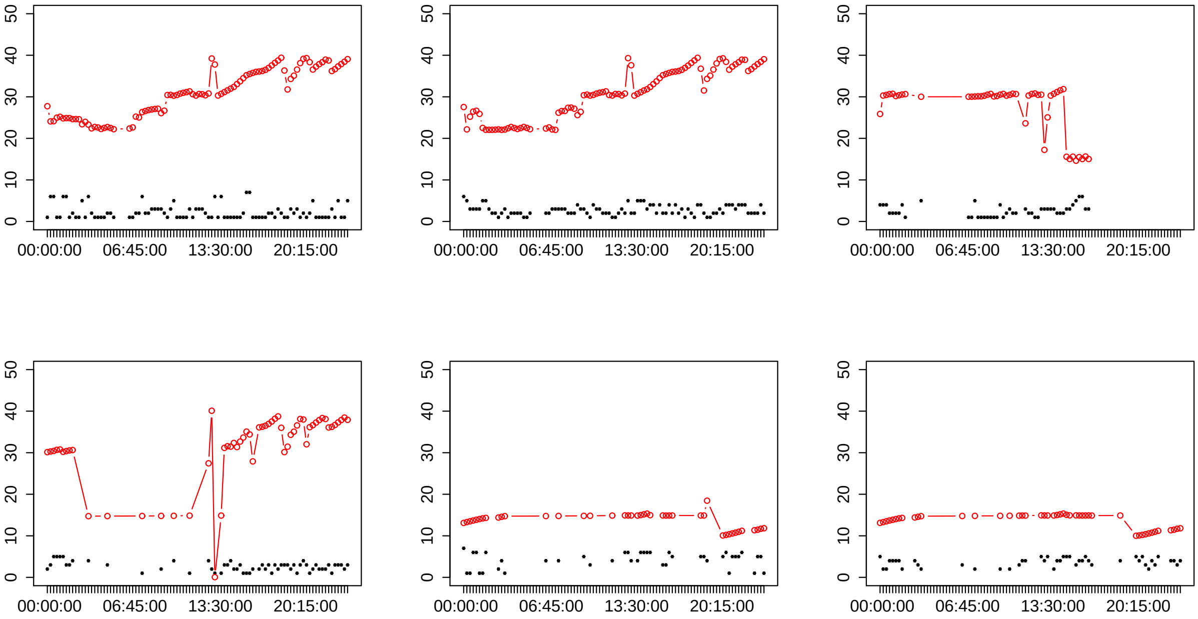


Figure 6: An illustration of bidding prices per user over one day for six bidders. The hollow circles are bidding prices, while the solid dots are the positions of the same bidder. It seems that the bidding prices are not closely correlated with the positions that the same user can achieve.

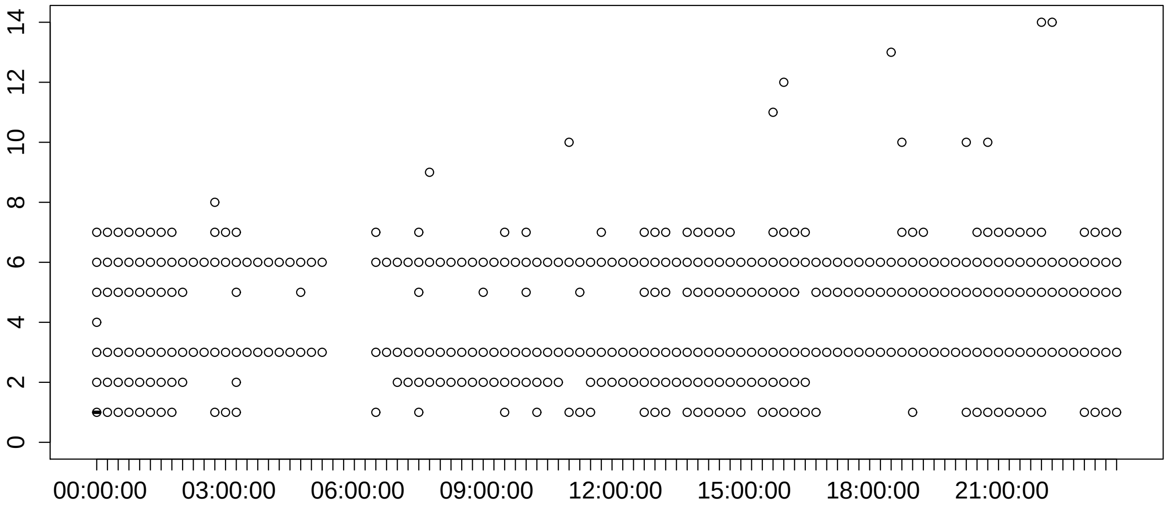


Figure 7: Bidding pattern of all 14 bidder on a phrase on November 15, 2002.

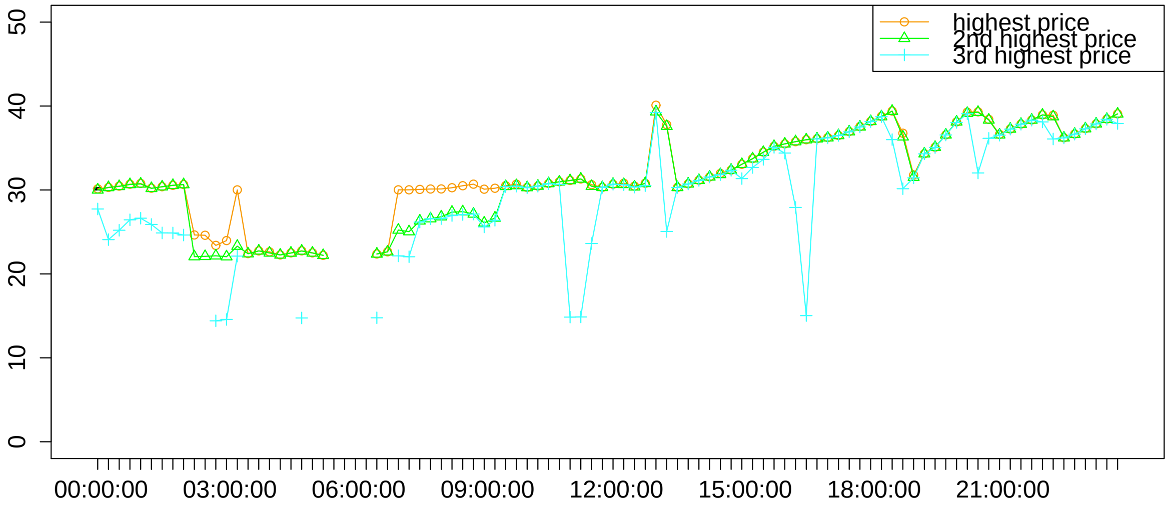


Figure 8: Highest three bidding prices over the time on November 15, 2002.

2 Game Theoretic Conditions for Equilibrium

To argue that (1) is a reasonable bidding scheme for a specific bidder at an equilibrium, let's assume that bidder j adopts the following scheme: she increase her bid by 1 cent per time until she reaches a point that her profit prospect goes down. We spell out what we mean by "profit prospect" shortly. If bidder j bids p_{tj} at time t and gets position k_{tj}, then her real profit should be $G_{k_{tj}j}(q_j - p_{tj})$. However, it is not convenient to use the real profit as an objective function for bidder to maximize, because otherwise, the bidder will always lower her bid to the price that is bidden by the bidder who occupies the next position (which was denoted by $b^{k_{tj}+1}$ earlier). Instead, the bidder consider her profit prospect, which is defined as $G_{k_{tj}j}(q_j - b^{k_{tj}+1})$, i.e., the maximal profit that this user can get without leaving her position. The optimal price (denoted by p^*) is defined to be the one such that when the bidder bids higher than p^*, her profit is reduced: i.e., if $p > p^*$, then we have

$$G_{k-1,j}(q_j - p) < G_{kj}(q_j - b^{k+1}),$$

where k is the short notation for k_{tj}. For the optimal price p^*, we then expect that the equality is achieved:

$$G_{k-1,j}(q_j - p^*) = G_{kj}(q_j - b^{k+1}).$$

Solving the above equation gives us (1).

We need to give an example to show that the above strategy does lead to an equilibrium. **This needs to be done....**

After all bidders intentionally or unintentionally bid according to the scheme in (1), assume that an equilibrium is reached, we would like to study the properties of these equilibriums. The following assumption is what we have observed in our numerical studies. Recall that we have denoted the number of clicks by bidder j for position k as G_{kj}. We will denote $\lambda_{kj} = \frac{G_{kj}}{G_{k-1,j}}$ as the percentage of clicks that bidder j would get if she moves down by one position from position k. We assume that this ratio λ_{kj} is no larger than 1 for all time t and bidder j.

Assumption 2.1. *For different bidder k, the ratio $\lambda_{kj} = \frac{G_{kj}}{G_{k-1,j}}$ does not depend on j and only depends on k. In other words, there are constants λ_k, such that $\forall k, j$, we have $\lambda_{kj} = \lambda_k$.*

At the same time, for a specific equilibrium, it will be called *output truthful*, if a bidder with higher expected return always takes a higher position in the equilibrium. We provides a formal definition as follows.

Definition 2.2. *Suppose at time t, an equilibrium has been reached. This equilibrium is* **output truthful** *if for any two different bidders i and j, if $q_i > q_j$ (i.e., bidder i has higher expected return per click), then $k_{ti} < k_{tj}$ (i.e., bidder i takes a higher position in the auction).*

Theorem 2.3. *If every bidder adopts the strategy that is described at the beginning of Section 2 to maximize their profit prospect, then when an equilibrium is achieved, such an equilibrium is output truthful.*

Proof. We use contradiction in our proof. We will argue that if the equilibrium is not output truthful, then one bidder does not maximize her profit prospect, which will contradict to the scheme that she is adopting. Recall that the essence of the embodied by (1) is that every bidder maximizes her profit prospect. If the output truthful property is violated, without loss of generality, we can assume that there are bidder i and j, we have $q_i < q_j$ and i and j take the k and $k+1$st position in the auction; i.e., a bidder with smaller expected return occupies a higher position. Since j takes the $k+1$st position, we have

$$p_{tj} = q_j - \frac{G_{k+1}}{G_k}(q_j - b^{k+2}). \tag{4}$$

Here G_{k+1} and G_k are the expected clicks for position $k+1$st and kth. Because of Assumption 2.1, we can easily show that G_k can be independent of bidders. Since bidder j occupies the $k+1$st position, we have

$$p_{tj} = b^{k+1}. \tag{5}$$

Now the following sequence of inequalities will reflects that for bidder i, its profit prospect in position $k+1$st is higher than its profit prospect in position kth, which contradicts the assumption that every bidder has maximized their profit prospect. We have

$$\begin{aligned}
\text{profit prospect for bidder } i \text{ at position } k &= G_k(q_i - b^{k+1}) \\
&\overset{(5)}{=} G_k(q_i - p_{tj}) \\
&\overset{(4)}{=} G_k(q_i - q_j + \frac{G_{k+1}}{G_k}(q_j - b^{k+2})) \\
&= G_k(q_i - q_j) + G_{k+1}(q_j - b^{k+2}) \\
&= (G_k - G_{k+1})(q_i - q_j) + G_{k+1}(q_i - q_j) + G_{k+1}(q_j - b^{k+2}) \\
&= (G_k - G_{k+1})(q_i - q_j) + G_{k+1}(q_i - b^{k+2}) \\
&< G_{k+1}(q_i - b^{k+2}) \\
&= \text{profit prospect for bidder } i \text{ at position } k+1.
\end{aligned}$$

The last inequality above is due to the assumption $q_i < q_j$. (Note that we have assumed earlier $G_k \geq G_{k+1}$ for all k.) □

Note that (1) is also similar to the forward-looking scheme in Bu et al. (2007) and Bu et al. (2008). We derived it under a slightly different scheme.

From the above, we can establish the uniqueness of the equilibrium.

Theorem 2.4 (Uniqueness)**.** *Under the same bidder strategy as in Theorem 2.3, when an equilibrium is achieved, it will be the unique equilibrium.*

Proof. The uniqueness is based on Theorem 2.3, which implies that the positions of bidders are determined by their expected returns, (consequently their relative positions are fixed,) plus the equation (1). □

Convergence of the aforementioned bidding scheme and the speed of convergence. This is a hard question...

3 Numerical Solution

The problem (3) can be reformulated as follows.

$$\begin{aligned}
\min_{q_j, \lambda_{kj}} \quad & \sum_{t,j} [p_{tj} - q_j + \lambda_{kj}(q_j - b^{k_{tj}+1})]^2, \\
\text{subject to:} \quad & 0 < q_j, \text{ and } 0 < \lambda_{kj} \leq 1.
\end{aligned} \tag{6}$$

Question: is the above a **convex** optimization problem?

Task: Derive Newton's iteration algorithm to solve for q_j and λ_{kj}.

3.1 Numerical Experiment

We solved the aforementioned problem with auction prices of keyword coded 2 on June 15, 2002. There are 12 advertisers who bade on this keyword. Their bidding prices are recorded every 15 minutes. Hence for each bidder, there are at most 96 prices. However, not all advertises bid all the time. We found that out of a 12 by 96 matrix, which has 1152 cells, 422 of them are nonempty. The following table gives a snapshot of the table from column 45 to column 50 (i.e., 11:00am to 12:30pm), which represent the normal pattern in this data set. Entry "-" indicates that the advertiser did not bid at that time interval.

Bidder	Interval 45	46	47	48	49	50
1	11.5700	11.7583	11.7675	11.9000	12.3520	12.3300
2	11.6020	11.8044	11.9775	12.1533	12.3663	12.5463
3	11.5777	11.7273	11.7905	11.9184	12.3167	12.4029
4	11.5500	-	-	12.2150	12.2650	12.6050
5	-	11.3000	-	-	-	-
6	11.5500	11.6533	-	11.6200	12.2567	-
7	11.6179	11.7963	11.8797	12.0964	12.3615	12.4913
8	-	-	-	-	-	-
9	11.0700	11.1850	10.3100	-	-	-
10	11.2600	-	-	-	-	-
11	11.7000	1.7000	-	-	-	-
12	-	3.1200	-	-	-	-

After solving the minimization problem in (6), below are the solutions to q_j and λ_{kj}. The computed expected returns, at a descending order, are:

Bidder ID	Estimated Expected Returns	Number of Bids (out of 96)
7	36.66	43
2	16.95	82
3	15.75	83
4	15.46	77
1	15.44	83
6	15.44	41
11	14.01	2
5	13.64	3
9	12.18	4
10	12.01	2
12	5.61	1

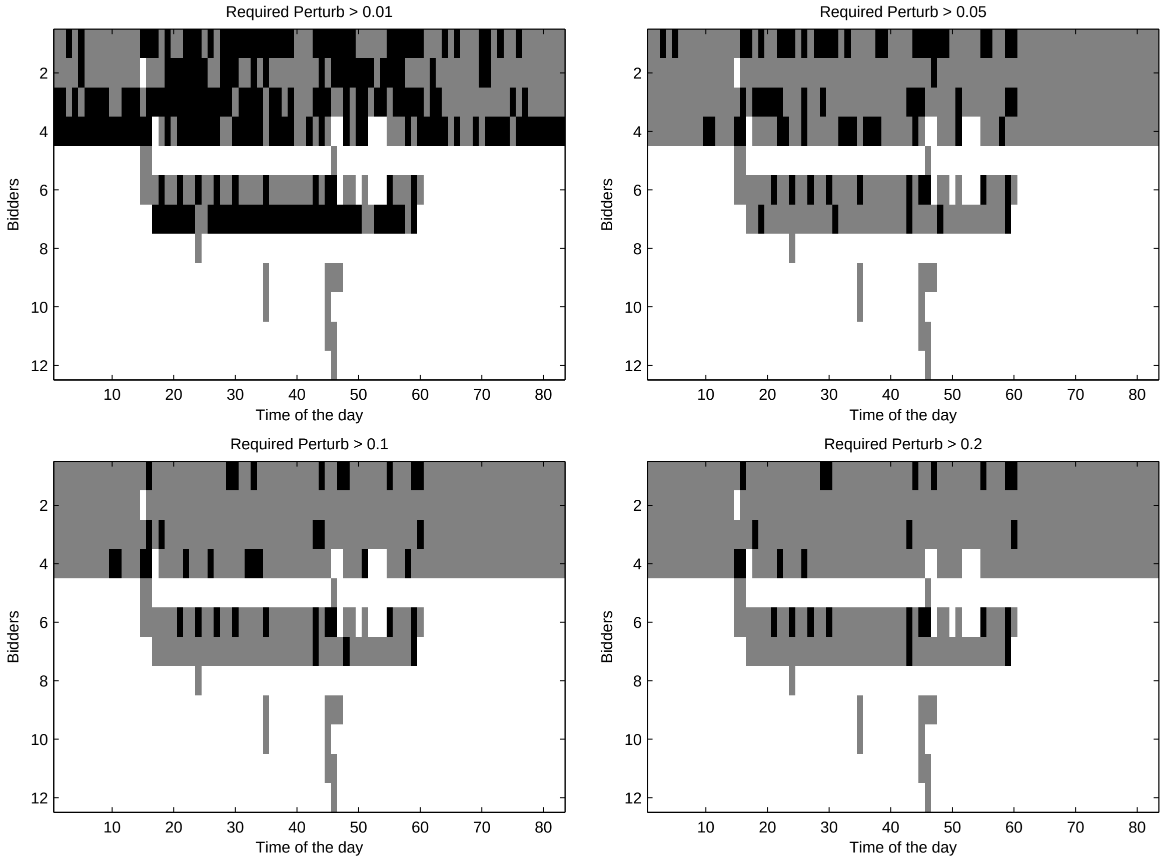
In the above table, the first column contains the identities of these bidders. The second column are the estimated expected returns. The last column are the total number of bids that each bidder made during the day. We can see that the last five bidders did not bid for more than 5 times during this day; consequently, the estimate of their associated expected returns and ratios (λ_{kj}) may not be meaningful, due to lack of observation points.

The following gives the computed λ_{kj}'s. If there is no data corresponding to certain cells, then the λ_{kj}'s can't be estimated; hence the cells are empty.

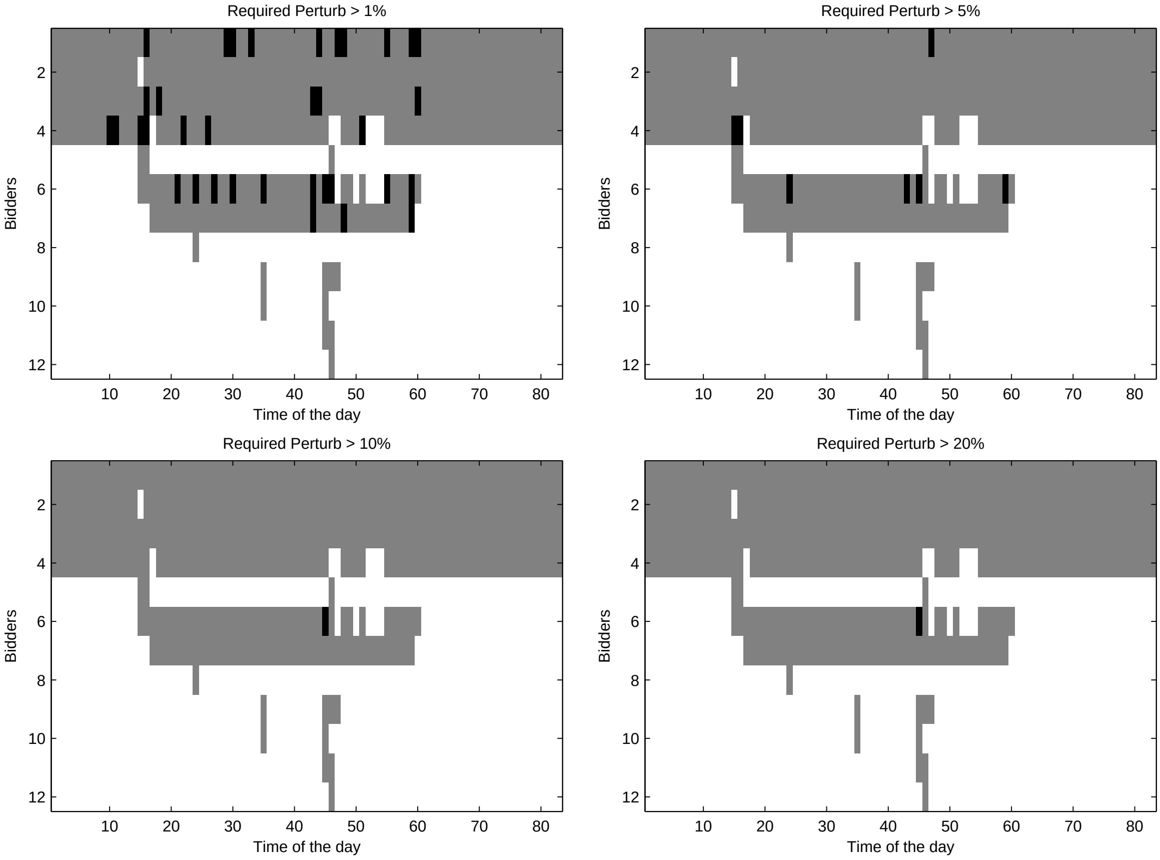
Bidder	Position 1	2	3	4	5	6	7	8
1	0.9900	0.9973	0.9659	0.8790	0.9696			
2	0.9961	0.9942	0.9957					
3		0.8860	0.9926	0.9824	0.9673	0.3645		
4	0.9910	0.9793	0.9258	0.9952	0.9965	0.9309		
5						0.9532		
6			0.9378	0.9932	0.8541	0.2698		
7	0.9994	0.9978	0.9980	0.9995				
8								
9							0.1102	
10							0.5505	0.7983
11	0.9656							
12								0.6365

It is interesting to see that many λ_{kj}'s are close to one, which indicates that moving one position up or down does not change the number of clicks significantly. Those λ_{kj}'s that are not close to one tend to occur when the associated bidders have not occupied many positions (e.g., bidders 9-12), or they correspond to the lowest positions for the bidders (e.g., $\lambda_{6,3}$ and $\lambda_{6,6}$). There is no estimated λ_{kj} for bidder 8, because there is no enough data.

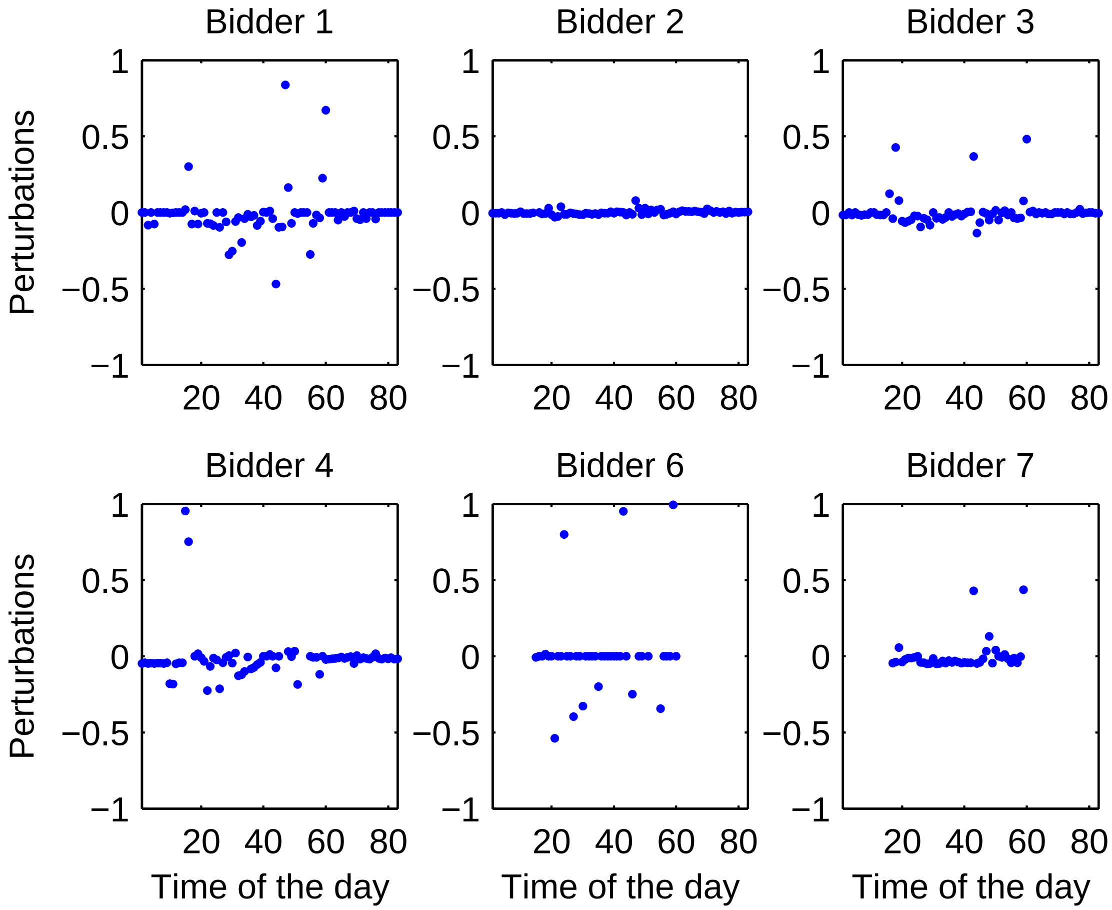
The values of the perturbation ε_{tj} indicate the goodness of fit of the proposed model. In the following graphs, we can observe the perturbation in absolute values.



The units are dollars. The white cells indicate that there is no bid from the corresponding bidders (which are rows) in the auctions (which are columns, or time of the day). Dark cell indicates that the absolute values of the perturbation is higher than the threshold that is displayed in the title of the figure. From the lower-right, one can conclude that most of the perturbations are no more than 20 cents.

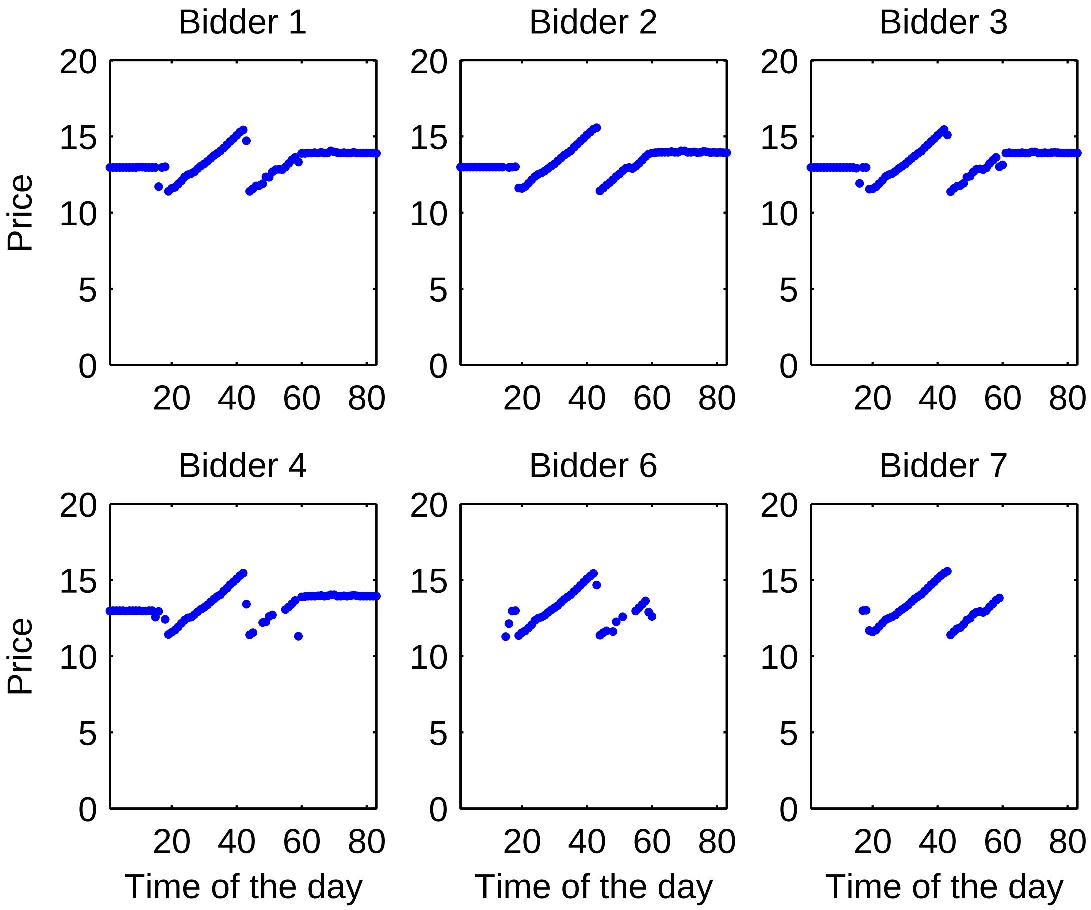


It is also informative to look at the perturbation in ratios: ε_{tj}/p_{tj}, where p_{tj} is the bidding price. From the above, we can tell that most of the perturbation is no more than 5% of the bidding price. A bidding price can range from 0 to 60 dollars. The maximal perturbation that we found is around 2 dollars.



Since there are not many bids from bidders 5 and then 9 thru 12, we decide to illustrate the perturbations ε_{tj} for bidders 1 thru 4, and then 6 and 7. The above figures display the perturbations in dollars. As we stated earlier, most of these perturbations are very close to zero.

It is easy to conjecture that the large perturbations that emerged in the last figures are due to abnormal bidding prices. It turns out that it is not true. The following figures displayed the bidding prices corresponding to the perturbations that were shown earlier. When the large perturbations occur, the corresponding bidding prices did not deviate much from the usual pattern of these bidders. We believe that these large perturbations are most likely caused by those rare bidders. They did not bid much; however when they join the auction, it has a significant impact on the approximation of the equilibrium model.



In conclusion, in this day and for this phrase, not much perturbation is needed in order to fit the model that is motivated by equilibria. This is unanticipated when we first started this project.

4 Discussion

connection of our approach to the work on latent variable in Rutz et al. (2012).

We may integrate the position modeling from Jerath et al. (2011).

References

Auerbach, J., Galenson, J., and Sundararajan, M. (2008), “An Empirical Analysis of Return on Investment Maximization in Sponsored Search Auctions,” in *ADKDD*, Las Vegas, Nevada, USA.

Bu, T.-M., Deng, X., and Qi, Q. (2007), “Dynamics of Strategic Manipulation in Ad-Words Auction,” in *WWW2007*, Banff, Canada.

— (2008), “Forward looking Nash equilibrium for keyword auction,” *Information Processing Letters*, 105, 41–46.

Edelman, B. and Ostrovsky, M. (2007), “Strategic bidder behavior in sponsored search auctions,” *Decision Support Systems*, 43, 192–198.

Edelman, B., Ostrovsky, M., and Schwarz, M. (2007), “Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords,” *American Economic Review*, 97, 242–259.

Edelman, B. and Schwarz, M. (2010), “Optimal Auction Design and Equilibrium Selection in Sponsored Search Auctions,” *American Economic Review*, 100, 597–602.

Jerath, K., Ma, L., Park, Y.-H., and Srinivasan, K. (2011), “A “Position Paradox” in Sponsored Search Auctions,” *Marketing Science*, 30, 612–627.

Rutz, O. J., Bucklin, R. E., and Sonnier, G. P. (2012), “A Latent Instrumental Variables Approach to Modeling Keyword Conversion in Paid Search Advertising,” *Journal of Marketing Research*, XLIX, 306–319.

Yuan, J. (2012), “Examining the Yahoo! Sponsored Search Auctions: A Regression Discontinuity Design Approach,” *International Journal of Economics and Finance*, 4, 139–151.