Exploratory and Interactive Daily Deals Recommendation

Anisio Lacerda[1] Adriano Veloso[1] Nivio Ziviani[1,2]

[1]Universidade Federal de Minas Gerais
Computer Science Department
Belo Horizonte, Brazil
{anisio, adrianov, nivio}@dcc.ufmg.br

[2]Zunnit Technologies
Belo Horizonte, Brazil
nivio@zunnit.com

ABSTRACT

Daily deals sites (DDSs), such as Groupon and LivingSocial, attract millions of customers in the hunt for products and services at significantly reduced prices. A typical approach to increase revenue is to send email messages featuring the deals of the day. Such daily messages, however, are usually not centered on the customers, instead, all registered users typically receive similar messages with almost the same deals. Traditional recommendation algorithms are innocuous in DDSs because: (i) most of the users are sporadic bargain hunters, and thus past preference data is extremely sparse, (ii) deals have a short living period, and thus data is extremely volatile, and (iii) user taste and interest may undergo temporal drifts. In order to address such particularly challenging scenario, we propose new algorithms for daily deals recommendation based on the explore-then-exploit strategy. Users are split into exploration and exploitation sets − in the exploration set the users receive non-personalized messages and a co-purchase network is updated with user feedback for purchases of the day, while in the exploitation set the updated network is used for recommending personalized messages for the remaining users. A thorough evaluation of our algorithms using real data obtained from a large daily deals website in Brazil in contrast to state-of-the-art recommendation algorithms show gains in precision ranging from 18% to 34%.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Filtering

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

Daily Deals, Recommendation, Exploration-Exploitation

1. INTRODUCTION

The daily deals business is a group-buying concept based on a synergistic view: helping small-businesses to advertise their products and services, while also helping customers to access significantly discounted offers (aka. deals). Typically, potential customers register as members of a daily-deals website and receive online offers and invitations, mostly by email. This concept has gained popularity over the last years, leading to myriad *daily-deals sites* (DDSs) and making web-based group-buying discount market extremely competitive. This forced top players, such as Groupon and LivingSocial, to shift their tatics towards providing more relevant and appealing deals, instead of non-personalized and ineffective email messages which are commonly viewed as spam.

In this paper we propose novel algorithms specifically devoted to daily-deals recommendation. The algorithms are based on the explore-then-exploit strategy. In our context, exploration is related to the "pursuit of information" that might come to be known, while exploitation is related to the "use" of the acquired information. Thus, we separate a fraction of the users to receive non-personalized messages (with non-biased deals) during the exploration phase. Rather than suggesting personalized deals, exploration focuses on capturing user interaction by gathering the most recent feedback (i.e., the purchases of the day). Then, in a posterior phase, feedback captured during exploration is exploited for suggesting personalized deals to the remaining users.

We employ a co-purchase network structure which evolves as users purchase deals. In this case, each node represents a user, and an edge between two nodes is created once the corresponding users purchase the same deal. Intuitively, a deal becomes more likely to be relevant to a particular user if he/she is close to users that have also purchased this deal recently.

We conducted a systematic evaluation involving real data obtained from PeixeUrbano,[1] the largest DDS in Brazil. The experiments showed that the proposed explore-then-exploit algorithms are extremely effective for daily-deals recommendation, providing precision improvements that range from 18% to 34%, when compared against existing recommendation algorithms.

The major contributions of this paper are:

- We propose explore-then-exploit algorithms and investigate the dilemma between exploration and exploitation under the daily-deals recommendation scenario.
- We propose criteria based on network centrality so that users are sorted in a way that increases the amount of feedback gathered during exploration, and also the amount of feedback indeed used during exploitation.
- We thoroughly evaluate the proposed explore-then-exploit algorithms using real data from a large DDS in contrast to state-of-the-art recommendation algorithms.

[1]www.peixeurbano.com.br

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
RecSys '13, October 12–16, 2013, Hong Kong, China.
Copyright © 2013 ACM 978-1-4503-2409-0/13/10 ...$15.00.
http://dx.doi.org/10.1145/2507157.2507228 .

The remainder of this paper is organized as follows. In Section 2, we review the related literature on DDSs and on exploration-exploitation recommendation strategies. In Section 3, we describe our explore-exploit algorithms. In Section 4, we describe the dataset, baselines, evaluation methodology and the results of the experimental evaluation of the explore-exploit algorithms. Finally, in Section 5, we present our concluding remarks.

2. RELATED WORK

The two most popular problems related to daily deals recommendation are: (i) deal-size estimation [3], and (ii) deal ordering for revenue maximization [7]. The first problem refers to the number of coupons that a DDS is expecting to sell for a given deal. The second algorithmic problem can be stated as: given a set of candidate deals, which are the ones that a DDS should feature as daily-deals in order to maximize its revenue. In both scenarios, the focus is to maximize the benefit of the DDS. In this sense, our work is closer related to [7].

Also related to our work are exploration-exploitation strategies applied to recommender systems [8] and related problems [11]. Usually, authors consider the multi-armed bandit setting [9, 8]. Under this setting, basically, we must simultaneously attempt to acquire new knowledge and optimize decisions based on existing knowledge. Therefore, it is necessary to maximize reward based on the knowledge already acquired, while attempting new actions to further increase knowledge. While we could have modeled the daily-deals recommendation problem under the bandit setting, we took other direction and propose alternate heustistics specifically devised to the problem.

3. EXPLORE-EXPLOIT ALGORITHMS

In this section we introduce algorithms for daily-deals recommendation based on the explore-then-exploit strategy [6]. Exploration is related to the pursuit of information that might come to be known, while exploitation is related to the use of the acquired information in order to improve a particular objective. More specifically, we take advantage from controling the order in which users receive email messages featuring the deals of the day. During the exploration phase, a subset of users receive non-personalized messages, and we acquire information regarding deals recently purchased by users that have a similar taste. We follow a $\epsilon-$first strategy, in which a pure exploration phase is followed by a pure exploitation phase. In this case, given a total of N users, the exploration phase occupies $\epsilon \times N$ users, while the exploitation phase occupies the remaining $(1-\epsilon) \times N$ users. Figure 1 describes the entire process of daily-deals recommendation based on the explore-then-exploit strategy. balancing exploration and exploitation.

3.1 The Exploration Phase

The exploration phase consists of picking a particular recommendation algorithm, and following the deals it suggests based on the past preference data collected so far. Thus, a fraction ϵ of the users will receive email messages featuring the k deals suggested by the recommendation algorithm. During this process users provide feedback when a suggested deal is indeed purchased, and this feedback is embedded into a structure we call *co-purchase network*.

The co-purchase network is an undirected and unweighted graph used to represent users that have purchased at least one deal in common, that is, nodes are users and an edge is created between two users if they have purchased a deal in common at least once. The co-purchase network makes evident users sharing similar tastes recently, and therefore it provides important information for the sake of personalized daily-deals recommendations. During the exploration phase, the co-purchase network is updated with the most recent purchases.

3.2 The Exploitation Phase

The exploitation phase consists of picking a particular recommendation algorithm, and following the deals it suggests based not only on past preference data, but also on current taste information. More specifically, we modified the recommendation algorithm so that it becomes aware of the proximity between users in the current co-purchase network. Then, we force the algorithm to weight more heavily deals that were recently purchased by nearby users. Thus, a fraction $(1-\epsilon)$ of the users will receive email messages featuring the k deals suggested by the recommendation algorithm, but these users are likely to benefit from more appealing and relevant deals that were recently purchased by users sharing similar tastes.

3.3 Splitting Users

At some point during the explore-then-exploit process, the set of users must be split into two partitions $-$ the partition composed of users $\{u_1, u_2, \ldots, u_{b-1}\}$ is devoted to exploration, while the partition composed of users $\{u_b, u_{b+1}, \ldots, u_N\}$ is devoted to exploitation. We may apply some criteria in order to sort users, and separate a fraction ϵ for exploration, and a fraction $(1-\epsilon)$ for exploitation. Ideally we should explore users that: (i) are more likely to provide feedback (i.e., to purchase a suggested deal), and (ii) share similar tastes with many other users. These two properties are more apparent in users that are more central in the co-purchase network. Therefore, we employ different centrality measures [4, 2] in order to sort users, as discussed next.

- Degree: given an arbitrary user in the co-purchase network, centrality is defined as the number of users having similar taste, that is, the number of edges focusing on the user.
- Betweeness: given an arbitrary user in the co-purchase network, centrality is defined as the number of times the user acts as a bridge along the shortest path between two other users in the network.
- PageRank: given an arbitrary user in the co-purchase network, centrality is defined as the number of all users that can be connected to the user through a path, while the contribution of a distant user is penalized by an attenuation factor.

Splitting by exploring central users. Our first splitting strategy directs most central users to receive recommendations during the exploration phase. Clearly, exploring central users has some advantages: (i) central users are likely to provide feedback, and (ii) many other users may share the same taste of central users.

4. EXPERIMENTAL EVALUATION

In this section we empirically analyze the recommendation performance of our explore-then-exploit algorithms. We employ MAP (mean average precision). All experiments were performed on a 1.93 GHz Core i7 machines with 8GB of memory.

4.1 Dataset

We employ real data obtained from PeixeUrbano, the largest DDS in Brazil. Our dataset comprises two months of information about deals, users and purchases. Specifically, during this 2-month period, 31,642 users purchased at least one out of 455 deals, resulting in a total number of 43,274 purchases. On average, each

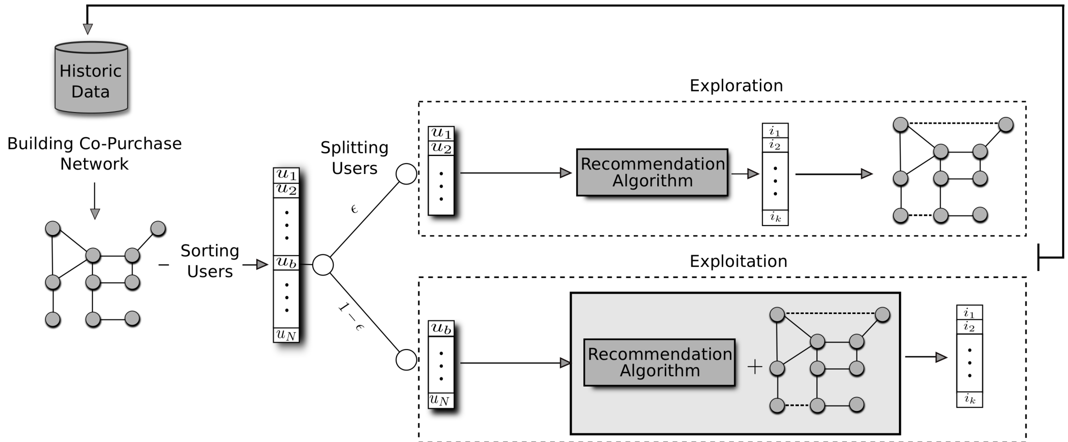

Figure 1: The daily-deals recommendation process. A fraction of the users are separated for exploration, and the co-purchase network is updated with the most recent purchases. The remaining users are benefited during exploitation. This process is repeated on a daily basis.

user purchases 1.36 deals, and thus the data is extremely sparse if compared with more studied recommendation scenarios, such as NetFlix (with 208 ratings per user) and MovieLens (with 166 ratings per user). Also, each deal remains valid, on average, for only 4 days.

4.2 Baselines

We used several recommendation algorithms as baselines. Most-Popular is a simple algorithm which recommends deals based on their popularity considering all previously purchased deals. In this case, the algorithm always suggests the same deal, no matter the target user. We also employ more sophisticated algorithms, including WRMF [5], a state-of-the-art matrix factorization algorithm, and BPR-MF [10], as a representative of collaborative filtering algorithms for binary relevance data.

4.3 Evaluation Methodology

Our evaluation follows the Interleaved Test-Then-Train methodology [1], in which each all recommendations user u_i receives on day t are evaluated, and then all deals purchased by user u_i on day t are included into the historical data available at day $t + 1$. The results to be reported correspond to an average over all days. Significance tests were performed (p<0.05) and the best results, including statistical ties, are shown in bold. Our evaluation takes three dimensions into account: (i) the fraction ϵ of users to be explored, (ii) the recommendation algorithm used during the process, (iii) the criterion used to sort users.

4.4 Results and Discussions

Our first experiment concerns the study of lower and upper bounds for the recommendation performance using the proposed explore-then-exploit strategy. Specifically, we executed 1,000 runs for each ϵ value, and in each run we randomly split users into exploration and exploitation. Therefore, we have 1,000 results for each ϵ value, from which we calculate the average, the best and the worst performance numbers for each ϵ value, as shown in Figure 2. The performance numbers shown in the figure will serve as lower and upper bounds for the recommendation performance.

The next experiment involves a detailed analysis of a very simple recommendation algorithm – the Most-Popular algorithm – when coupled with the proposed explore-then-exploit strategy. Table 1 shows performance numbers in terms of MAP. The analysis

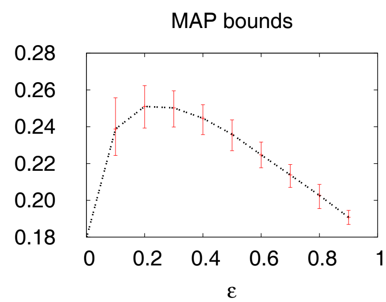

Figure 2: (Color online) Users were randomly split into exploration and exploitation (1,000 runs). Empirical lower and upper are derived from the worst and best results, respectively.

consists in evaluating the different sorting network-centrality criteria. The best performance numbers with each sorting criteria are underlined, and the best overall numbers are shown in bold. The first row, for which $\epsilon = 0$, shows the performance numbers obtained with no exploration, and this is equivalent to running the original Most-Popular algorithm (i.e., with no feedback information), and thus we consider the first row as baseline numbers. In terms of MAP, improvements over the baseline range from 29% to 34%. The recommendation performance of the Most-Popular algorithm is always close (and many times surpasses) the empirical upper bound shown in Figure 2, no matter the evaluation measure employed. In almost all cases, sorting users according to their betweeness leads to higher improvements over the baseline.

Figure 3 shows MAP numbers for the WRMF algorithm (top) and BP-RMF algorithm (bottom). Each plot shows the MAP performance during the exploration, exploitation and overall (i.e., the entire process) phases. The plots also show MAP numbers obtained by Most-Popular (symbol •) and the (empirical) upper bound for MAP (symbol ▲). From the plots, the trade-off between exploration and exploitation is clear, and the best value for ϵ is usually around 0.3 (i.e., the best overall performance). Notice that Most-Popular is the best performer in all cases, no matter the criterion used to sort users. Algorithms WRMF and BP-RMF show similar

ϵ	Degree Explore	Betweeness Explore	PageRank Explore
0.00	0.199 •	0.199 •	0.199 •
0.01	0.201 ↑	0.214 ↑	0.203 ↑
0.05	0.202 ↑	0.224 ↑	0.204 ↑
0.10	0.206 ↑	0.240 ↑	0.213 ↑
0.20	0.213 ↑	0.258 ↑	0.222 ↑
0.30	0.224 ↑	**0.264** ↑	0.237 ↑
0.40	0.231 ↑	0.255 ↓	0.237 •
0.50	0.228 ↓	0.252 ↓	0.240 ↑
0.60	0.227 ↓	0.244 ↓	0.235 ↓
0.70	0.224 ↓	0.233 ↓	0.232 ↓
0.80	0.213 ↓	0.221 ↓	0.221 ↓
0.90	0.204 ↓	0.207 ↓	0.205 ↓
avg.	7.5%	17.6%	10.6%

Table 1: MAP numbers, considering the Most-Popular algorithm. Symbol ↑ indicates that the corresponding result is an improvement over the result obtained using the previous ϵ value. The last row shows the average improvement considering all ϵ values.

MAP numbers, with WRMF being slightly superior in most of the cases.

5. CONCLUSIONS

This paper focused on the important problem of suggesting relevant and appealing products and services to potential customers, considering the particularly challenging scenario of daily-deals recommendation. Specifically, we consider the task of sending personalized email messages featuring potentialy relevant deals to users. In this case, we can impose a restriction on the order that users receive their messages, that is, some users receive their messages before others. We propose explore-then-exploit recommendation algorithms that are devised to: (i) gather feedback from users that receive their messages first (i.e., during the exploration phase), and (ii) use the gathered feedback in order to send personalized messages to the remaining users (i.e., during the exploitation phase).

There is a trade-off between exploration and exploitation, in the sense that the more users are explored, more feedback is gathered but less feedback is indeed used. To deal with this trade-off, we propose sorting users according to their centrality in an evolving co-purchase network, so that more central users are those that share the same taste with more users. We employ a 2-month real data snapshot obtained from PeixeUrbano, the largest daily-deals website in Brazil, and show that: (i) the recommendation performance of existing algorithms is no better than the naive strategy of suggesting non-personalized deals for all users (i.e., all users receive the same message featuring the same deals), and (ii) the proposed explore-then-exploit algorithms are very effective and well-suited to the daily-deals recommendation scenario, providing improvements ranging from 18% to 34%.

Acknowledgments

This work is partially supported by the National Institute of Science and Technology for the Web, MCT/CNPq grant 57.3871/2008-6, and by the authors' individual grants and scholarships from CAPES and CNPq. We thank PeixeUrbano for providing the data used in the experiments.

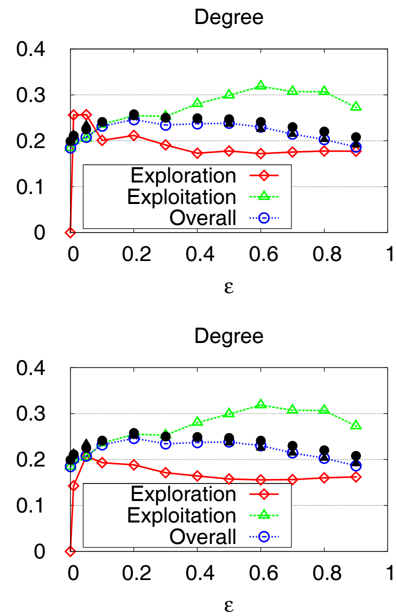

Figure 3: (Color online) MAP numbers obtained considering: (i) only the exploration phase (red curve), (ii) only the exploitation phase (green curve), and (iii) overall (blue curve). Symbol • indicates the corresponding MAP numbers for Most-Popular. Symbol ▲ indicates the (empirical) upper bound for MAP. The plot at the top corresponds to the WRMF algorithm and the plot at the bottom corresponds to the BP-RMF algorithm.

6. REFERENCES

[1] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer. MOA: Massive online analysis. *Journal of Machine Learning Research*, 11:1601–1604, 2010.

[2] P. Bonacich. Power and centrality:a family of measures. *The American Journal of Sociology*, 92(5):1170–1182, 1987.

[3] J. Byers, M. Mitzenmacher, and G. Zervas. Daily deals: prediction, social diffusion, and reputational ramifications. In *WSDM*, pages 543–552, 2012.

[4] L. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1(3):215–239, 1979.

[5] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.

[6] L. Kaelbling, M. Littman, and A. Moore. Reinforcement learning: A survey. *J. Artif. Intell. Res.*, 4:237–285, 1996.

[7] T. Lappas and E. Terzi. Daily-deal selection for revenue maximization. In *CIKM*, pages 565–574, 2012.

[8] L. Li, W. Chu, J. Langford, and R. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, pages 661–670, 2010.

[9] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *ICML*, pages 784–791, 2008.

[10] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[11] Z. Xu, K. Kersting, and T. Joachims. Fast active exploration for link-based preference learning using gaussian processes. In *ECML/PKDD (3)*, pages 499–514, 2010.