Contextual Recommender Problems

[Extended Abstract]

Omid Madani
madani@yahoo-inc.com

Dennis DeCoste
decosted@yahoo-inc.com

Yahoo! Research
74 N. Pasadena Ave, 3rd floor
Pasadena, CA 91103

ABSTRACT

The contextual recommender task is the problem of making useful offers, *e.g.*, placing ads or related links on a web page, based on the context information, *e.g.*, contents of the page and information about the user visiting, and information on the available alternatives, *i.e.*, the advertisements or relevant links. In the case of ads for example, the goal is to select ads that result in high click rates, where the (ad) click rate is some unknown function of the attributes of the context and ad. We describe the task and make connections to related problems including recommender and multi-armed bandit problems.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning - Induction

General Terms

Algorithms

Keywords

Recommenders, Multi-Armed Bandit, Personalization, Exploration-Exploitation, Regression, Reinforcement Learning, Data Mining, Utility

1. INTRODUCTION

Users (browsers) select pages to view and the task is to put one or more ads on such pages. The contextual (ad) problem consists of making such selection and placement decisions in order to maximize the expected return over some period of time, where expected return is a function of the likelihood of the ads being clicked – and possibly even a transaction or purchase taking place – and the prices of those clicked ads. Information that may significantly aid such decisions include page and user attributes, such as content and site information and users' recent behavior, and attributes of the available ads, such as ad content and bid prices. The horizon or time period for optimization will also implicitly or explicitly figure into the problem. However, a major challenge is *sparsity*: there may be many ads available (*e.g.*, millions), whereas the number of interactions we may get from a single typical user, in a time period of interest, may be very small in comparison (*e.g.*, a handful a day), and furthermore click rates for arbitrary ads are relatively small as well (*e.g.*, one percent).

We explore a number of different ways of viewing the problem. These viewpoints reveal the different aspects of the task, or may just reflect the type of available resources and data. While we focus on the task of selecting ads to show (*contextual adevertising*), we expect that the abstraction, the *contextual recommendation problem*, applies to other tasks such as (personalized) web search and web page organization (see Section 7). Our focus in this paper is on an informal exploration. We leave fomralizations and concrete solutions to future work.

The paper is organized as follows. Section 2 describes the information that should be useful. It identifies an important distinction: the system has some control over choice of ads but does not have control over choice of users or, in general, contexts. Section 3 discusses the contextual problem as a prediction problem, ignoring the controllable (decision theoretic) aspects of the problem. Section 4 discusses a simplified version of the contextual task as a standard n-armed (multi-armed) bandit problem. Section 5 extends the n-armed bandit viewpoint and explains connections with typical recommender problems. The number of users of the system can be in millions and the collaborative or the "community" aspects of the problem can help significantly in addressing sparsity and making better decisions. Section 6 in turn combines the problems of Sections 3 and 5 and describes perhaps the most general problem in which information about contexts and ads are represented as points in large dimensional spaces, but the decision theoretic, multi-armed bandit, and community aspects may all be taken into account for better performance. The contextual task is a challenging problem in which increasing utility is a direct function of improving algorithms.

2. THE AVAILABLE INFORMATION

In this paper, for simplicity, we assume that the objective is to maximize ad click rates. Two major types of informa-

UBDM '05 , August 21, 2005, Chicago, Illinois, USA.

tion that we may have to make optimal decisions are: (1) the *context attributes*, such as information about the user and site, and (2) the information about the arms, *e.g.*, information about the ads, which we refer to as *arm attributes*, such as ad topics and bid prices. The distinction is that we (the decision maker) do not have control over the choice of context attributes but we can choose from among the arms and therefore we have some control over the choice of arms and their attribute values.

Attributes (features) of the overall *context*, include page attributes, such as: page content, page topics, site, source, time and date. Context also includes attributes of the user, such as: user id, demographics, overall interests, and recent searches. Attributes of the arms (mostly ads) include: click price, overall click rate, ad contents, advertiser account id, ad topics, ad url and contents of the page pointed to. We may also have some control over the presentation of the page with the ads, placement of the ads, highlighting, and so on. Therefore, the list of potentially predictive features (both for contexts as well as arms) is long. These features may have different feature types (numeric, boolean, probabilistic, ..). Furthermore, not all the attribute values may be available at all times (missing values). The ideal system handles all these possibilities effectively.

Depending on the exact problem formulation, we may need other information. For example in the Bayesian formulation of the problem, we need priors over ad click rates (given context and arm attributes), and the horizon we want to optimize over [2]. Finally, while we don't have control over context attributes, we may know something about the distribution of what we will encounter.

3. THE PREDICTION PROBLEM

Assume we had access to a function that would predict the click rate well given the context and arm information. Such a function would be able to handle a large number of features, potentially with many missing values. Given a sufficiently large matrix of feature values and click outcomes, we could learn such a function via various algorithms (linear methods, decision trees, knn, ..). This problem is basically a typical machine learning problem (prediction, regression, etc). However, an issue is how the training data is produced or how to obtain such data. The context of feature values should be representative of the distribution of contexts we get.[1] But another potentially bigger issue is the choice of arm(s) for each context. This is under our control, and the question is how to make such decisions. There are several alternatives to choosing arms, including:

1. Uniformly at random (pure exploration). This may be fine if we don't have too many arms or possible arm feature values and/or we have a significant amount of time to explore, without much concern for exploitation.

2. We have strong beliefs/priors that only a limited number of arm values are relevant for each context. Then problem reduces to a series of smaller explore-exploit problems, and we can apply the appropriate strategy (*e.g.*, random sampling of 1) in this case.

3. (dynamic) Experiment design: in this case, we want to be more selective in the choice of arms, depending on the contexts, in order to increase our accumulated rewards while we are learning more about the world (exploring). A number of methods, ranging from standard multi-armed bandit to optimization algorithms such as genetic algorithms could be explored. See section 4.

Note that we may require not just the click rate (an expectation), but ideally a distribution to be output from such a function, in order to capture the level of uncertainty of the function over the predicted expectation. In practice, we expect that the training matrix will always be relatively limited compared to the space of possible feature values, and therefore significant uncertainties would remain.

Given that we have such a predictor function, we could use it to obtain the best action (choice of arm) given a context as well as where to explore. We may also use it to identify the patterns (combinations) of context and arm features that lead to relatively high click rates.

This pure learning approach, ignoring the controllable aspects of the problem, is very similar to the work of Joachims who explored learning better ranking functions using click data in the context of web search [4].

4. THE MULTI-ARMED BANDIT PROBLEM

As it is clear from previous discussion the overall problem involves exploration and exploitation at some level. Exploration means: to explore different arms (*e.g.*, ad types) to better estimate click rates and more effectively find winner arms, and exploitation means: to choose to pull those arms that are currently known to yield good rates, and exploit their good rate of returns. One could treat each context individually, without taking into account information about other contexts. With this simplification the problem becomes a standard multi-armed bandit problem [2]. There is significant literature on the problem: there are a number of ways of formulating the problem, and a number of algorithms and heuristics exist, with a substantial understanding of many theoretical and empirical properties of these technqiues [6, 1, 9, 3, 2].

Consider learning for a single user, *i.e.*, the user (user id) is our context. Also, assume the arms are ad topics. When a user visits a page, the systems task is to pick a certain ad topic, and from that ad topic pick a certain ad to show.[2] The objective is to maximize click rate over some period of time. We could then initialize the arm priors (say in a Bayesian formulation of the n-armed bandit problem), and figure out which arms work best for that person. A major problem we face with this approach is the problem of *sparsity*: there may be many ad topics available (thousands and beyond), whereas the number of interactions we may get from a single user, in a time period of interest, may be very small in comparison. The average baseline click rate (when showing an arbitrary ad) is very low (*e.g.*, below one percent). In standard n-armed bandit problems, the scale, *i.e.*, the number of arms is much more limited and/or the number of interactions or desired horizon needs to be significantly larger for descent optimization opportunity [3].

[1]However exceptions exist, for example when we are trying to focus on some subset of the feature space or when active learning.

[2]In general, we assume most individual ads appear and disappear too quickly to obtain sufficient statistics.

[3]A subtle difference with the typical n-armed bandit prob-

When there is no information differentiating the arms, there is no way around the sparsity problem. However, often we have much information that has the potential to better focus the experimentation and lead to better returns (click rates). Such information may be obtained from similar users and/or demographics information on that user. Additionally, the arms are not independent. Some ads are closer to one another than others along some dimensions (eg college football ads are closer to college basketball ads than to political ads with respect to the topic aspect). Therefore the arms are not independent and the information obtained for one arm can affect what we know about the other arms. In a Bayesian setting, such information influences the priors over the click rates. Still, as the population of users interact with the system we obtain more information about user behavior and potential user and arm similarities that can further help the choice of displayed arms. In the next section, we further develop the idea of using similar contexts and arms to affect decisions more dynamically.

5. THE RECOMMENDER PROBLEM

Consider a matrix where the rows are users (user ids) and the columns are ad topics. Whenever a user visits a page an ad topic is picked (via some mechanism) and shown to the user, and the outcome is recorded (whether or not there was a click). We expect that with a sufficiently large population of users and collection of ad topics, many users would behave similarly, and cluster into a smaller number of groups. Similar behavior, in our case, means similar patterns of likelihoods of clicking on certain topics. Such clustering also imposes clusters on the ads (columns). Thus click through rates that are known for some users can be used to infer similar click through rates for other similar users. The similarity among users may be defined in terms of click through rates themselves and/or inferred/predicted at least partly based on other user attributes such as their demographics and recent and past behavior. In the same vein, similarity among the items (columns) may be a function of the meta attributes of topics (*e.g.*, similarity in terms of subject) in addition to the click through rates the topics obtain from the users' activity.

Therefore, when we want to select ads to display to a single user, treating the problem as merely a single narmed bandit task misses much opportunity for faster optimization. This problem is very similar to recommendation problems [8], and involves many similar issues: users (in general, explicit contexts) with similar tastes, missing values, and our choice of the columns/items to show when a user (re)visits. Some difference from typical recommender problems are:

1. In recommendation problems, one does not recommend the same item again after it has been viewed or bought, but here, items/arms (*e.g.*, , a certain ad topic) can be repeatedly shown so better estimates of click rates or expected revenue for it is obtained.[4] The missing quantity is the rate (more generally a distribution) rather than grade of likability.
2. In many recommender problems, users have some control of what they see and rate, here the control is completely under the systems from the outset.[5]
3. Often in a recommendation problem there is no explicit notion of reward. Here, time lost is revenue lost, and exploration/exploitation and reward maximization take a markedly more relevant role.

Several recommender solutions methods, in particular collaborative filtering approaches, as well as techniques such as dimensionality reduction and clustering, nearest neighbors and other machine learning methods, apply to the contextual problem as well. Perhaps the most important distinguishing factor is the very direct connection of superior algorithms, *e.g.*, better clusterings or similarity metrics, to higher returns. In case of clustering, this could be contrasted with standard applications of clusterings in which the true objective may be ill defined or subjective. In contextual problems, the ultimate objective is the expected returns, and the performance of any context and arm clustering would be measured against that ideal of how effectively it increases expected returns. See for example the work of Kleinberg et. al. [5], which studies algorithms for clustering for similar economic end goals. The challenge here is how to do exploration and exploitation with the understanding that information obtained about a single user can help the whole community of users, and information about the community can help better serve a single user. Similar questions apply to the arms.

6. THE GENERAL PROBLEM

When the set of possible contexts and arms are enumerable, techniques developed for the problems of previous section are directly applicable, and sampling methods may address scalability. This "manageable" scenario occurs when, for example, contexts are restricted to user ids and arms are restricted to ad topics (see Section 5). This assumption may turn out to be too restrictive in practice. To allow for the full power of prediction, each context or arm can have a number of attributes "active" (*e.g.*, a page may belong to multiple topics), where the space of possible attributes can range in 100s and beyond. Thus, both a context and an arm may be viewed as points in their own large dimensional spaces.

In such scenarios, the set of possible contexts and/or arms is not enumerable and cannot be represented explicitly. And yet, all the aspects of the problem, the prediction problem, the exploration vs exploitation problem, and the recommender problem remain. Ignoring any aspect may lead to significant loss of opportunity for optimization. Obviously, techniques in learning and optimization in large dimensional spaces, such as dimensionality reduction and learning similarity metrics, is relevant. However, we are not aware of prior research that addresses exploration and exploitation in such large dimensional spaces, taking community (collaborative filtering effects) into account.

lem is the perspective of optimization: often the problem is cast from the user's point of view. Thus the user is presumably motivated to interact and experiment actively and intelligently to recognize and exploit the better arms. In the case of contextual problems, the system (the company) is primarily doing the optimization, though the users should benefit as well.

[4]However, in our problem there may also be a notion of decay in interest. This in part depends on how we define the context.

[5]Still it is possible to imagine scenarios where we allow the users to pick the type of ads they want to see.

7. SUMMARY

We described the contextual recommender problem, motivated by contextual advertising, and identified several related subproblems:

- A prediction problem involving a large number of features, possibly with missing values. The objective is to obtain a predictor that outputs a distribution preferably instead of a single numeric quantity.
- A generalization of the multi-armed bandit problem to a set of contexts, in order to address sparsity issues. This problem may also be considered a special recommender problem and, in particular, collaborative filtering techniques are relevant.
- Further extension to the case where the space of context or arms is not enumerable: A context or an arm is modeled as a point in a large dimensional feature space.

There may remain other challenges, for example nonstationarity: A user's needs and interests change over time. Contextual recommender problems are general. For instance, instead of ads, other items can be offered, for example navigational links [7]. Personalized web search may also be viewed as a special case in which knowledge of the query significantly reduces ambiguity and the need for extensive exploration [4]. Studying these problems in theory as well as developing experience and an understanding of effective practical algorithms should be of great value.

8. REFERENCES

[1] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The non-stochastic multi-armed bandit problem. *SIAM Journal on Computing*, 2002.

[2] D. A. Berry and B. Fristedt. *Bandit Problems.* Chapman and Hall, 1985.

[3] J. Gittins. *Multi-Armed Bandit Allocation Indices*. John Wiley and Sons, 1989.

[4] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining.* ACM, 2002.

[5] J. Kleingberg, C. Papadimitriou, and P. Raghavan. Segmentation problems. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 1998.

[6] O. Madani, D. J. Lizotte, and R. Greiner. The budgeted multi-armed bandit problem. In *COLT*, 2004.

[7] M. Perkowitz and O. Etzioni. Towards adaptable web sites: Conceptual framework and case study. *Artificial Intelligence*, 2001.

[8] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3), 1997.

[9] R. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT press, 1998.