A penny for your thoughts? The value of information in recommendation systems

Alexandre Passos
Computer Science Department
UMass Amherst
apassos@cs.umass.edu

Jurgen Van Gael
Microsoft Research Cambridge
jurgen.vangael@gmail.com

Ralf Herbrich
Microsoft Research Cambridge
rherbrich@googlemail.com

Ulrich Paquet
Microsoft Research Cambridge
ulripa@microsoft.com

1 Introduction

Most recommendation systems are trained to predict behavioral data and then used to generate more such data by recommending items and receiving feedback on the quality of these recommendations. This data in then fed back into the training process. This creates a feedback loop: as long as the low-cost way to interact with the service is through the recommender, the recommender will only ever see behavioral data on the items it chooses. This process can lead to hidden biases, as it effectively limits how much information the recommender system will ever see. On the other hand, there is a cost to making exploratory recommendations, as they should, myopically, be worse than the best bets of a recommendation system. In this paper we explore the notion that recommender systems are a special kind of active learning agents, with the peculiarity that the cost of asking for the label of an instance depends on its true label, as the cost of showing a bad recommendation when exploring is higher than the cost of showing a good recommendation.

This raises an important question: how much should a recommendation system explore, and when? In this paper we attempt to answer this question by looking at the information value of a rating, i.e., the expected gain in future recommendation quality after knowing the rating a user would assign to an item. As computing this value exactly requires strong assumptions about the scenario in which the system operates and is computationally intractable, we show how to efficiently compute lower bounds to the information value of a set of recommendations and how to use these lower bounds to choose sets of recommendations that are better than those selected greedily.

The key assumption in this paper is that the recommendation process, as far as the system and a single user are concerned, can be accurately modeled with an expectmax game [5, 8], where the recommendation system maximizes utility and the user acts according to a probability distribution. If we allow the system to have prior beliefs over the actions of a user—more specifically, hierarchical priors learned from the behaviors of other users—then it is possible to compute the expected quality of future recommendations given that a user will rate any given set of items. This knowledge then can be used to tradeoff between exploration and exploitation.

This approach naturally attacks some of the main problems of recommendation systems:

- **The cold-start problem**: traditionally, there are many approaches to improve the recommendations for users who haven't rated items by using feature information; while it is very difficult to make accurate predictions in the absence of rating data it should be possible to make good recommendations in the absence of certain predictions, by considering the information that will be obtained from the resulting feedback.

- **Overpersonalization**: in most recommendation systems all it takes is a small number of "likes" and the system becomes heavily biased towards items similar to those a user has already seen and rated, as these will necessarily have a high expected reward. When using this information value approach the system will always also consider the possible future rewards in case the user likes a given set of items that has not being shown so far. This will avoid the worse effects of overpersonalization since while some uncertainty remains about the user's tastes the system will explore to try to achieve better future recommendations.
- **Diverse sets of recommendations**: even assuming a recommendation system manages to make correct predictions about the probability of the user liking certain items, simply recommending greedily will lead to homogeneous sets of recommentations, where for example all action movies are ranked above all drama movies even though there is a fair chance a user might actually want a drama movie. As the conditional information value is often higher for dissimilar items this behavior is avoided as long as uncertainty remains in the model.

2 A family of policies based on the information value

Consider a two-player game in which one player acts on randomly and the other acts to maximize a utility function. There are two possible orderings of events, as far as the utility player is concerned. Either the utility player acts before knowing the action of the expectation player, and hence, on average, his reward will be $\max E[R]$, the maximal expected reward; or the utility player acts after knowing the action of the other player, and hence, on average, his reward will be $E[\max R]$, the expected maximum reward.

The expected reward of the case where the utility player acts in ignorance is a lower bound on the reward of the full information case, or $\max E[R] \leq E[\max R]$, with equality only when the action taken by the expectation player cannot possibly affect the outcome of the game. To see this, note that both sides of the inequality can be written as maximizing a sum of functions, where the left-hand side has an equality constraint and in the right-hand side each term of this sum can use the optimal value of the free variable. One can also see this by using Jensen's inequality and the fact that the supremum of a set of linear functions is a convex function.

In this paper we consider that the quality of a recommendation system is the quality of the recommendations it actually makes (rather than the quality of its predictions of the ratings, or other common measures). Assuming star-rating data, or like-dislike data, then, we define the true reward function $R(S)$ of a set of recommendations S to be the number of stars or the number of likes of items in that set.

One of the differences between recommendation systems and standard bandit scenarios is that it is unfeasible to expect that any given user will rate even a small fraction of the whole collection of items. Assuming that we know a priori that a user is likely to return to the system N times, and that at an access the system is allowed to recommend K items to that user, the quality of the recommendation system's recommendations for that user, a posteriori, is then just the sum of the qualities of each set of recommendations the system has made to the user.

The value of such a game, and the optimal recommendations at each round, can then be computed as follows. Assuming binary rewards, and with $R(S)$ being the utility of set S of recommendations, $L(S)$ the set of possible feedback values one can get on the set S, and $P(L(S)=l)$ the probability that the set S is labeled according to l, the optimal set of first recommendations and the value of the game are

$$S = \arg \max_{S:|S|=K} E[R(S)] + \sum_{l \in L(S)} P(L(S)=l) \max_{S':|S'|=K} \left\{ E[R(S')|L(S)=l] + \sum_{l' \in L(S')} \dots \right\}, \tag{1}$$

where implicit in the $\dots$ is a recursive insertion of that same equation N times. This is similar to the Gittins index, except it is defined here on sets of items rather than actions, and that we do not make the binomial assumption (rather using a black-box predictor of $P(R|S)$, which can condition on anything) that makes computing the Gittins index computationally tractable.

Even using dynamic programming this is only computable in exponential time due to searching over an exponential number of labelings. Hence we suggest that we focus here on the myopic one-step approximation to this true information value,

$$S = \arg \max_{S:|S|=K} E[R(S)] + \sum_{l \in L(S)} P(L(S)=l) \max_{S_i:|S_i|=K} \left\{ \sum_{i=2}^{N} E[R(S_i)|L(S)=l] \right\}, \quad (2)$$

essentially replacing the sum over all future possibilities to a one-level scoop down the branching tree. The inner sum is effectively the expected score of the next N recommendations, after conditioning on observing the rewards of the current recommendations. As noted above, this objective function is a lower bound of the true value of the game. This "information value" objective function generalizes what is usually done in recommendation systems by (1) adding a future lookahead and (2) using a tighter lower bound on the future expected rewards of the actions of the recommendation system.

It is possible to trade off the tightness of the lower bound and the computational complexity of the optimization problem as one wishes by carefully replacing $E[\max R]$ terms with $\max E[R]$ terms in equation (1).

As the assumption that one knows for sure how many times each user will return to the recommendation system is unrealistic, we propose treating N as a hyperparameter to be tuned with usage data and adding another α hyperparameter to discount the value of future reward, thus forming the objective function used in the remainder of this paper:

$$S = \arg \max_{S:|S|=K} E[R(S)] + \alpha \sum_{l \in L(S)} P(L(S)=l) \max_{S_i:|S_i|=K} \left\{ \sum_{i=2}^{N} E[R(S_i)|L(S)=l] \right\}. \quad (3)$$

To optimize this we use greedy search and sampling, and instead of searching over all possible items we search only over an n-best list computed greedily.

3 Experiments

Here we evaluate these policies using two practical experiments: a simulation experiment based on completing the data and a user study.

3.1 The simulation experiment

In the reinforcement learning literature it is common to design simulators to test different policies in the lab without performing expensive live experiments [19]. Our simulation-based experiment first completes an observed rating matrix using collaborative filtering and then plays the recommendation scenario many times with different policies. The main flaw of this type of experiment is that it is unrealistic, and it is impossible to tell whether the simulation is implicitly favoring one family of policies over another.

For this experiment we use the MovieLens 100K dataset [6] and the Matchbox [16] online bayesian collaborative filtering algorithm. We split the list of users in the dataset in two groups, one with the 45 most prolific raters and the other with all the remaining users. We train a Matchbox recommender on the full dataset, and set it aside to use as ground truth. Then we train another matchbox recommender on the group of least prolific users to get the right prior for users and items in this context. Finally, for each of the more prolific users, we use a copy of the less traind model to recommend them movies with each poslicy, assuming the predictions of the held out model are the truth. This is done for 20 rounds of 10 recommendations. We report, for each policy, the average over users of the star rating of the recommended movies. We used 30 latent dimensions for the trait vectors of Matchbox.

The recommendation policies we considered are:

- **greedy**: Always recommend items with higher predicted rating.
- **random**: Always recommend items randomly.

Policy	Average reward
greedy	4.59
random	3.77
ϵ-**greedy**-0.1	3.78
ϵ-**greedy**-0.5	3.78
UCB-0.1	4.59
UCB-0.5	4.59
UCB-1	4.55
VI-1-50	4.70
VI-0.1-50	4.69
VI-0.01-50	**4.71**
VI-1.0-100	4.70
VI-0.1-100	4.69
VI-0.01-100	**4.71**

(a) The average star value of items recommended in the simulation experiment.

Movie type	Information value	Greedy
New	$\mathbf{0.392} \pm 0.006$	0.354 ± 0.005
Familiar	$\mathbf{0.839} \pm 0.002$	0.823 ± 0.002

(b) 95% confidence intervals for the proportions of liked and desired movies in the user study.

- ϵ-**greedy**-α: With probability α follow **random**, otherwise follow **greedy**.
- **UCB**-α: use a policy based on LinUCB [10], maximizing $E[r] + \alpha\sigma$, where σ is the model predicted standard deviation of the reward distribution for each item.
- **VI**-α-N: The information value policy from equation (3)with linear discount factor α and looking at N items in the future to evaluate recommendations.

Table 1a shows the average star value of items recommended according to each policy in this experiment. The best values were obtained by the information value approaches—a result insensitive to hyperparameter settings.

3.2 A user survey

While very useful for testing and design of the policy, the simulation-based experiments are fundamentally unconvincing as, if no exploration was done when collecting the training data, it is unreasonable to expect that the simulation is well-behaved in the face of explicit exploration. For this reason we designed a user survey.

The survey was designed as a web page. Each user was sequentially presented with 10 rounds of 5 movie recommendations. For each of these recommendations the user marked whether they had "seen and liked", "seen and disliked", "not seen but want to see", or "not seen and don't want to see" the movie. The movies presented in each round will depend on the movies presented in the previous rounds and on the user feedback returned. For this experiment both the "seen and liked" and "not seen but want to see" were treated as identical positive feedback to the recommender, with all other options being treated as negative feedback. We again used the Movielens 100K dataset and a Matchbox model trained on the ratings in that dataset as the recommendation system. In this experiment we did not use any user or movie features.

The users were randomly divided in two groups: a control group which received recommendations according to a greedy policy and a treatment group which received recommendations according to a value-of-information policy. The survey was online for a period of about 24 hours in which it was taken by 67 people, among friends and colleagues of the authors.

Table 1b shows results of this user study. We report two separate ratios: the laplace-smoothed ratio of liked movies to all movies the user has seen and the laplace-smoothed ratio of movies the user wants to see to all movies the user hasn't seen. In both cases the information value group liked statistically significantly more movies, demonstrating that recommendations based on the information vale are on average better than greedy recommendations.

4 Conclusions and future work

In this paper we presented a family of policies for recommendation systems that elegantly balance between exploration and exploitation using the information value criterion as a guide. While very

expensive to compute exactly, a simple approximation allow making better recommendations than would be possible following either a greedy policy or simple exploration policies.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by UPenn NSF medium IIS-0803847. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

The authors would like to thank Thore Graepel and David Stern for helpful discussions while researching this paper, and David Soergel for substantial comments on the text.

References

[1] P. Auer, N. Cesa-Bianchi, Y. Freund, and R.E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2003.

[2] R.M. Bell and Y. Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[3] C. Boutilier and R.S. Zemel. Online queries for collaborative filtering. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*. Citeseer, 2003.

[4] C. Boutilier, R.S. Zemel, and B. Marlin. Active collaborative filtering. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, pages 98–106. Citeseer, 2003.

[5] U. Chajewska, D. Koller, and R. Parr. Making rational decisions using adaptive utility elicitation. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 363–369. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2000.

[6] J.L. Herlocker, J.A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM, 1999.

[7] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[8] R.A. Howard. Information value theory. *Systems Science and Cybernetics, IEEE Transactions on*, 2(1):22 –26, aug. 1966.

[9] A. Krause and C. Guestrin. Near-optimal nonmyopic value of information in graphical models. *Uncertainty in Artificial Intelligence UAI05*, 1(3.1):3–4, 2005.

[10] L. Li, W. Chu, J. Langford, and R.E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[11] B.M. Marlin and R.S. Zemel. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the third ACM conference on Recommender systems*, pages 5–12. ACM, 2009.

[12] S. Prasad. Using social networks to improve movie rating predictions.

[13] F. Radlinski and T. Joachims. Active exploration for learning rankings from clickthrough data. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 570–579. ACM, 2007.

[14] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, pages 784–791. ACM, 2008.

[15] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 441–448. Citeseer, 2001.

[16] D.H. Stern, R. Herbrich, and T. Graepel. Matchbox: large scale online bayesian recommendations. In *Proceedings of the 18th international conference on World wide web*, pages 111–120. ACM, 2009.

[17] A. Strehl, J. Langford, S. Kakade, and L. Li. Learning from logged implicit exploration data. *Arxiv preprint arXiv:1003.0120*, 2010.

[18] A. Strehl, J. Langford, S. Kakade, and L. Li. Learning from logged implicit exploration data. In *Arxiv preprint arXiv:1003.0120*, 2010.

[19] R.S. Sutton and A.G. Barto. *Reinforcement learning*, volume 9. MIT Press, 1998.

[20] P. Viappiani and C. Boutilier. Optimal bayesian recommendation sets and myopically optimal choice query sets. *Neural Information Processing Systems (NIPS). MIT press*, 2010.