Novel Algorithms for Ranking and Suggesting True Popular Items

Shaik Jameer[1], Syed Sadat Ali[2]

[1]*M. Tech, Nimra College of Engineering & Technology, Vijayawada, A.P., India.*
[2]*Assoc. Professor & Head, Dept.of CSE, Nimra College of Engineering & Technology, Vijayawada, A.P., India.*

Abstract: *Ranking is the process of giving rank scores to the most popular item by taking user feedback. The most frequently occurring items are given the highest rank score. In practice, one may use prior information about the item popularity. For example, in the survey, the user may select the suggested item or they may also select the others. Suggestion is a list of items that are presented to the users. This is done based on the user's feedback. The users give their preference of items through feedback and use them in the ranking of items. In this paper, our aim is to propose novel algorithms for suggesting popular items to users in a way that enables learning of the users' true preference over items. The true preference refers to the preference over the items that would be observed from the users' selections over items without exposure to any suggestions.*

Keywords: *M2S, PROP, Ranking, Suggestions.*

I. INTRODUCTION

Ranking is the process of giving rank scores to the most popular item by taking user feedback. The most frequently occurring items are given the highest rank score. We focus on the ranking of items where the only available information is the observed selection of such items. In learning of the user's preference over items, one may leverage some side information about the items, but this is out of the scope of this paper. In practice, one may use prior information about the item popularity. For example, in the survey the user may select the suggested item or they may also select the others. If they selected the already suggested items they will become more popular and if he does not they may get out of the popular list.

Suggestion is the list of items presented to the users. This is made based on the feedback of the user. The users give their preference of items through feedback and use them in the ranking of items. The main goal of this paper is to learn the true popularity of items and suggest them to the user. Item mentioned here can be anything like documents, files, search query keywords etc. A more specific application of this system is that of tagging process where items are tags applied to the content e.g. photo (in fickr), web pages (in delicious) and video (in youtube) etc.. The users can choose the appropriate tags for the information object based on their preference. The previous tagging system is based on the history of tagging. Figure 1 shows an example user interface to enter tags for a web page, for example, tagging system in BBC. Suggested items and most popular items are also provided. Users can select those items from suggestion or popular sets or create own tag items.

Figure 1: An example tag entry user interface

Suggestion of items to the users becomes complicated process in the popularity of items. The user tends to select such items from the suggested list more frequently. It is because of (1) Bandwagon (the user conform the choice of other users) (2) least effort (selecting from the suggested list is easier than to think another alternative) (3) Conformance in vocabulary (no need to write whole word accurately or correctly). So the suggestion can skew the popularity over the items [1]. The item "news" becomes more popular if that item is suggested frequently. We see that suggesting popular items creates some problems in the popularity of the items list, then why we made such suggestion? There are many reasons; say it recalls what the candidate items are. In this paper, our aim is to prepare some algorithms for ranking and suggesting so that it enables to learn the users' true preference over the items. The true preference is the user preference over the items without any exposure to any suggestions.

II. RELATED WORK

The problem studied in this paper relates to the broad area of some recommendation systems [2] in which the goal is to learn which items are preferred by the users based on the user's selection of items. Another related area is that of the voting systems. Specifically, our system could formally be seen as an instance of the approval voting [3] in that each user can select any set of candidates offered on a voting ballot. Our work is related to statistical learning problems of the multi-armed bandit type [4]). We consider a finite list of items. Each user is presented with an item that is selected by this user with unknown probability specific to this item.

An asymptotically optimal rule to decide which item to present was found in [4] and was further extended in [5] to allow presenting more than one item. In [6], the authors studied the entrenchment problem where the search engine result sets lock down to a set of popular URLs and proposed to intervene the results with randomly sampled URLs. In [7], the authors provide various statistical characterization results on the tagging in the social bookmarking application del.icio.us. In [8], the authors studied the effect of the tag suggestions on the users' choice of tags in MovieLens systems, which they instrumented for experiments. In [9], the authors provides an estimation procedure of the imitation rate defined in this paper and estimates for tagging of Web pages scenario.

III. PROPOSED WORK

A. Naïve Algorithm

Let us consider how the user selects the items. User selects an item from the entire list of items by sampling, using the true item popularity distribution *r*. Where

r= (r1, r2, ………., rc)

be the Users' true preference over the given set of items C and r called true popularity rank scores**.** Otherwise, user does get the same but confines his choice to items in the suggest set. The naive algorithm "TOP" which suggests a fixed number of the popular items, fails to determine the true popularity ranking of the items if the imitation probability in the user's choice model is adequately large. This can be explained in the Algorithm 1.

Algorithm 1: TOP (Top Popular)
Step 1: Init ci = 0 for each i item
Step 2: If item i is selected :
Step 3: ci ← ci +1
Step 4: S ← a set of s items with largest c values

Ranking process is made on the number of selection of items in the past. If an item "i" is selected then its count i.e., ci is incremented by one. Initially all the items count ci = 0 for each item "I". The suggestion set "S" contains s items.

B. Frequency Proportional Algorithm

Frequency proportional(PROP) is a randomized algorithm that for each user presents a suggestion set of items, sampled with probability proportional to the sum of the current rank scores of items. Also that this algorithm is computationally demanding when the number of items and suggestion set size s are non small; it requires sampling on a set of elements as shown in Algorithm 2.

Algorithm 2: PROP (Frequency Proportional)
Step 1: At the k- th item selsction
Step 2: Sample a set S of s items with probability

$$\propto \sum_{j \in S} \rho_j$$

C. Move- to- Set Algorithm

Move- to- Set(M2S) is a random iterative update rule of the suggestion set of items ,where the suggestion set is updated only when the user selects an item that is not in the suggestion set presented to the user. This algorithm suggests that the last used item for the suggestion set size of one item which is a recommendation rule used by many user interface designs. Due to the random eviction of the items from the suggestion set, M2S is different from suggesting the last distinct used items for the suggestion set size greater than one item although the rule prefers recently used items. As an aside, note that M2S algorithm relates to the self-organized sorting of items known as move-to-front heuristic as proposed in Algorithm 3.

Algorithm 3: M2S (Move- to- Set)
Step 1: At pth item selection
Step 2: If i item is selected and i is not in suggestion set, S
Step 3: Randomly remove an item from S
Step 4: Add i to S

D. Frequency M2S Algorithm

For each item, this algorithm keeps a counter of how many users selected this item over users that were not suggested this item. The rationale is not to update the counter for the items that were suggested and selected by users in order to mitigate the positive reinforcement due to exposure in the suggestion set. Furthermore, a selected item that was not suggested does not immediately qualify for the entry in the suggestion set (as with M2S), but only if its counter exceeds that of an item that is already in the suggestion set as in Algorithm 4. In addition, specific to FM2S is that the eviction of an item from the suggestion list is over a subset of items with smallest counter.

Algorithm 4: FM2S 9Frequency Move- to- Set)
Step 1: Init: Ni :=0 for each item i
Step 2: At pth item selection
Step 3: If i is selected and i not in S
Step 4: Ni ← Ni + 1
Step 5: If Ni greater than any N values of items in S
Step 6: Randomly remove one item from S
Step 7: Add i to S

Both M2S and FM2S learn true popularity ranking that are lightweight. Self tuning in that they do not require any special configuration parameters. FM2S algorithm confines to displaying only sufficiently popular items as the suggestion set can displayed as shown in the Figure 2.

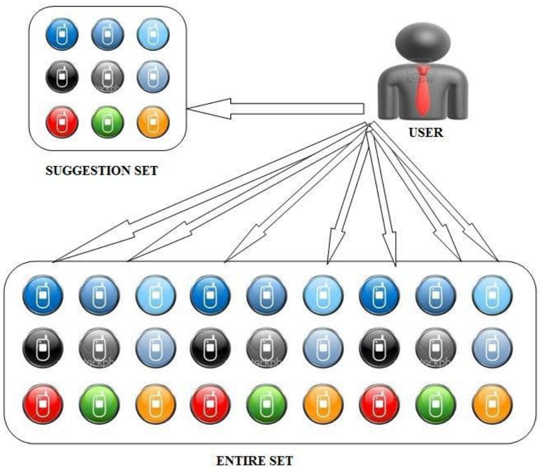

Figure 2: The Proposed Suggestion Set to the user.

IV. CONCLUSION

In this paper , to suggest the most popular items based on ranking process and popularity of items, we proposed the randomized algorithms like naive, PROP, M2S, FM2S.We assessed quality of suggestions which are measured by true popularity of suggested items, and we identified how limit the ranking of items are related to true popularity ranking. By learning the true popularity ranking of the items, the proposed objective of suggesting true popular items can be quickly achieved by using the proposed algorithms.

REFERENCES

[1] H.A. Simon, “Bandwagon and Underdog Effects and the Possibility of Election Predictions,” Public Opinion Quarterly, vol. 18, pp. 245-253, 1954.
[2] R. Kumar, P. Rajagopalan, and A. Tomkins, “Recommendation Systems: A Probabilistic Analysis,” Proc. 39th Ann. Symp. Foundations of Computer Science (FOCS), 1998.
[3] http://en.wikipedia.org/wiki/Approval_voting, 2009.
[4] T.L. Lai and H. Robbins, “Asymptotically Efficient Adaptive Allocation Rules,” Advances in Applied Math., vol. 6, pp. 4-25, 1985.
[5] V. Anantharam, P. Varaiya, and J. Walrand, “Asymptotically Efficient Allocation Rules for the Multiarmed Bandit Problem with Multiple Plays—Part i: i.i.d. Rewards,” IEEE Trans. Automatic Control, vol. 32, no. 11, pp. 968-976, Nov. 1987.
[6] S. Pandey, S. Roy, C. Olston, J. Cho, and S. Chakrabarti, “Shuffling Stacked Deck: The Case for Partially Randomized Ranking of Search Engine Results,” Proc. 31st Int’l Conf. Very Large Data Bases (VLDB), 2005.
[7] S. Golder and B.A. Huberman, “The Structure of Collaborative Tagging Systems,” J. Information Science, vol. 32, no. 2, pp. 198-208, 2006.
[8] S. Sen, S.K. Lam, A.-M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F.M. Harper, and J. Riedl, “Tagging, Communities, Vocabulary, Evolution,” Proc. 2006 20th Anniversary Conf. Computer Supported CooperativeWork (CSCW), 2006.
[9] F. Suchanek, M. Vojnovi_c, and D. Gunawardena, “Social Tagging: Meaning and Suggestions,” Proc. 17th ACM Conf. Information and Knowledge Management (CIKM), Oct. 2008.