Temporal Context for Location-based and Preference-Aware Recommendation

Nhat Bui
nbui@ucla.edu

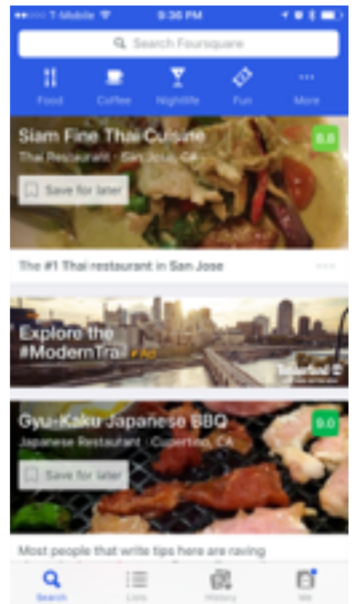
***Abstract*—The purpose of a recommendation system in a location-based social network is to present users with venues that are appealing to visit. In this paper, I present a multi-armed bandit built on top of a collaborative-filter that provides temporal-context to the recommendation system. This recommendation system will suggest venues that match a user's personal interest while considering time of day. The motivating factor is that there are venues that are more meaningful at certain hours of the day. The arms of the bandit are the preference-categories a user may indicate. Given a time of day, the bandit will suggest a category that will maximize value to the user. I will evaluate the performance of the system with a large dataset of user-location history from Foursquare. The results reveal that temporal context does not provide significant value to recommendations since venues do not display significant popularity at times of the day relative to other venues.**

Keywords*—*location-based social networks, location-based services, user preferences, recommendation systems, UCB1 bandit, exploration/exploitation

I. Introduction

Most location-based recommendation systems do not consider the temporal context of queries for recommendations. Typical location-based recommendation systems use collaborative-filtering to identify venues matching a user's personal interest. In these systems, the likelihood of a user visiting a venue is based on the rating the user has given a location. High ratings mean a user enjoyed their visit to a location whereas low ratings means a user did not enjoy their visit. High ratings imply that a user is more likely to visit the venue again and conversely, low ratings imply that a user is less likely to visit the venue again. Location ratings are inherently sparse since users can only visit a limited number of venues and users do not visit all venues in their geographic region [1]. Collaborative-filtering (CF) considers the ratings of other users to predict another user's ratings. In many systems, the similarity between different users is computed to determine how to weigh ratings of other users. In the system proposed by Bao et. al., a category hierarchy is used to determine other users with similar preferences [1]. Thus, their system consists of identifying other users with high similarity and then venues that will yield a high-rating among similar users are recommended.

In the system described above, the recommendations are based on geographic location and user preferences. Some venues have more popularity at different times of the day. For example, a venue with a mostly breakfast menu will most likely be popular in the earlier parts of the day whereas movie theaters will most likely be more popular in the evening. To this end, I implement an additional module to the recommendation system that considers the time of day in respect to recommendations. The module is modeled as a multi-armed bandit whose arms are the categories of the category hierarchy. The bandit implements the Upper Confidence Boundary 1 algorithm (UCB1) and is trained on a set of over 500,000 user check-ins on Foursquare for Tokyo, Japan from April 2012 to February 2013.



Constructing the rewards is challenging. A bandit whose arms are all possible venues in a geographic region is impractical. The preference-categories are the arms of the bandit.

My findings can be summarized as follows:

- Based on the values of the bandit's arms after training on user location histories, the use of categories as the arms create noisy values and are unreliable in distinguishing further value in recommendations.
- Temporal patterns for venue categories certainly exist but their patterns are mostly similar. They don't provide great distinction of a preference for a category at a certain time of the time relative to another.

II. System Description

This section describes the original recommendation system implemented by Bao, Zheng, and Mokbel. Then it presents the multi-armed bandit.

A. *System Overview*

The collaborative filter and the multi-armed bandit process user queries in parallel. Results from the collaborative filter are not inputs into the multi-armed bandit or vice versa. The chosen category (arm) of the bandit is used to re-evaluate the recommendations. The details of this will be described later when the structure of the rewards is introduced.

This architecture is in contrast with Matikainen et. al who built a multi-armed bandit on top of a CF but used the recommendations from the CF as the set of arms. In this model, the bandit only tracks the number of times each arm is pulled and exploration/exploitation is determined based on the values the CF provides. Though this implementation is promising, simulating it was impractical in this application. The simulation is performed over a large set of days. To keep true to the application, the offline part of the CF, to be described in the next section, must be executed per day in the simulation. It is a very expensive process and makes the simulation impractical. Because the bandit is independent of the collaborative filter in this case, it can be trained independently from the collaborative filter. Training offline has been evaluated to be reliable [3].

B. *Location-based Recommendation System*

The recommendation system can best be described as performing 2 parts: an offline and an online process. In the offline process, all user check-in data is evaluated in a batch process to determine local experts of categories and a user's preferences. In the online process, the system takes a user query and computes experts with high similarity to the user to return a set of recommendations with predicted high ratings. The offline processes uses a lot of techniques from web-search to determine category experts and user preferences.

To identify local experts of a location category, matrices of user-locations are created from users' histories. Specifically, users are the rows and venues are the columns of this matrix. Elements in the matrix are the number of times the venue has been visited by the user. Using a Hypertext Induced Topic Search (HITS) model, the hub score of a user is computed [4]. Experts have a high hub score in their category. In the HITS-model, the quality of venue, its authority score, is based on the number of visits to the venue. But not all visits are equal. Visits are weighed by the visitor's hub score. But a user's hub score is a function of visits to high-quality venues or high authority scores.

$$u_c.h = \sum_{u.v \in c} v_c.a$$

$$v_c.a = \sum_{u \in U} u_c.h$$

Thus, the problem is iterative and the formulated as an eigenvector problem, where the hub score of every user is the eigenvector to the user-location matrix. Using the power iteration method, the hub scores are computed for all users. Users with higher hub scores are potential candidates for local experts in the category.

$$\mathcal{A}_n = M^T \cdot M \cdot \mathcal{A}_{n-1}$$

$$\mathcal{H}_n = M \cdot M^T \cdot \mathcal{H}_{n-1}$$

The next offline process determines users' personal preferences. The process uses Term Frequency-Inverse Document Frequency (TF-IDF), another technique from web search, to determine preferences. A user's location history is regarded a document and categories are terms in the document [1]. This technique helps normalize the count of visits amongst categories in order to identify user preferences based on a statistically-significant number of visits. An example of this application can be shown from the domain it originated. A document, such as "Harry Potter and the Sorcerer's Stone" by J.K. Rowling, will exhibit a high term frequency for the word "the". On it's own, that would probably suggest some significance to the word or the author's writing style. However upon examining other documents, books in this case, for the term frequency of "the", it will probably reveal that Rowling's use of "the" is not significant as the term also displays similar frequencies in other documents. In the context of location-based social networks, some categories such as restaurants will exhibit more visits than museums. Thus, a user's preference is identified if they visit a category significantly more relative to the population and their own location history. The user's TF-IDF value is the user's category weight. A higher weight means the user's has a stronger preference for the category.

$$u.w_{c'} = \frac{|\{u.v_i : v_i.c = c'\}|}{|u.V|} \times \log \frac{|U|}{|\{u_j : c' \in u_j.C\}|},$$

The first online process selects experts based on how similar their preferences are. The calculation of similarity has two parts. In the first part, similarity is defined as the weighted sum of preferences, the TF-IDF value, between a user and another user. Thus, the more overlapped, heavily-weighted categories a user shares with another user indicates high similarity.

However, overlapped nodes is insufficient. The notion of entropy, H, is introduce to describe how concentrated someone's interests are.

With level similarity and entropy, the overall similarity can be determined between users.

$$LevelSim(u,u',l) = \sum_{c \in C^l} \min(u.w_c, u'.w_c)$$

$$H(u,l) = -\sum_{c \in C^l} u.P(c) \times \log u.P(c)$$

$$Sim(u,u') = \sum_{l=1}^{|l|} \beta \times \frac{LevelSim(u,u',l)}{1 + |H(u,l) - H(u',l)|}$$

In the second part of the online calculation, all similarities are multiplied with the count of visits to venues. The top *n* highest scoring venues are the recommendations.

$$R_u(v) = \sum_{u' \in \mathcal{E} \,\&\, v \in V} Sim(u,u') \times v(u',v)$$

C. *Temporal Multi-Armed Bandit*

The temporal multi-armed bandit is implemented as a Upper-Confidence Boundary bandit, specifically the UCB1 algorithm. Each arm of the bandit is a category. These categories are classifications of the venues and the hierarchy determines of the specificity of the classification. For example, a Chinese restaurant has the obvious category of "Chinese Restaurant" but resides within the parent category of "Food and Drink" which can include Mexican Restaurants, Italian Restaurants and much more. A user's preferences consist of categories that the user has expressed interest in. The bandit tracks the number of the times each arm is pulled and the value of each arm. However these are tracked in hourly bins. Specifically, one hour of the day will have its own count and values for each arm and another hour of the day will have its own count and values for each arm. In this way, there are virtually twenty-four bandits (this implementation chooses to divide the day hourly).

Algorithm: Temporal-Aware UCB1
Input: hour of day [1,23]
Output: A recommended category.

```
function select_arm(hr)
    max_v = 0
    best_arm = 0
    for arm in array[hr].arms
        v' = arm.v + sqrt(2*log(total_count)/arm.c)
        if v' > max_v
            max_v = v'
            best_arm = arm
    return best_arm

function update(chosen_arm, hr, reward)
    chosen_arm.c += 1
    total_count += 1
    n = chosen_arm.c
    chosen_arm.v = ((n-1)/n)*chosen_arm.v + (1/n)*reward
```

The UCB1 bandit uses the term below to dictate exploration and exploitation.

$$arm'.value = arm.value + \sqrt{2\frac{\log(\sum_{i \in arms} arm_i.value)}{arm.count}}$$

Rewards are determined by the level distance to the lowest common ancestor of the bandit's pulled category and the category of the actual category of the venue visited. In this approach, the category hierarchy is observed as a tree and levels are defined by the depth (distance from the root) of a category in the tree. Nodes of a level are further distinguished by their parent node to avoid the conclusion that distance from the root implies logical similarities between categories. For example, because water parks and nail salons are at the same distance from the root does not mean that users of one will more likely enjoy the other. One belongs to the parent node "Theme Parks" and the other belongs to "Health & Spa".

The level distance is computed by a recursive algorithm that starts at the deeper category and walks the tree until the common ancestor is found. So if the bandit recommends a "Basketball Court" at 4 PM and the user actually visited a "Botanical Garden", the level distance is two because the lowest common ancestor, "Outdoors & Recreation", is a level distance of two from the furthest category, "Basketball Court".

Because a UCB1 bandit is meant to maximize the rewards and increasing level distance represents more dissimilar categories, the following formula is used to compute rewards as a function of level distance. In this formula, increasing level distances reduces the reward. If the lowest common ancestor of two categories is the root of the hierarchy, there is no reward.

$$reward = 1 - \left(\frac{level\_distance}{number\_of\_levels}\right)^2$$

The pulled arm is interpreted as the recommended category for the hour and all recommended venues have their score multiplied by a factor determined by their categorical distance from the recommended category. It will be revealed that the multi-armed bandit does not significantly improve the recommendations and actually worsens performance.

The dependence of the bandit on time may imply that the bandit is actually a contextual bandit since the time does alter the value the bandit computes. However this is not the case because time is only used to index to a set of counts and values. A typical contextual bandit would use time as a dependent variable of a function to compute the potential value of each arm.

III. Simulation Results

The dataset consists of user histories of Foursquare check-ins for Tokyo, Japan from April 2012 to February 2013 [5]. Check-ins are valuable because it represents a user actually visiting a venue. However, rewards are implied because a check-in is interpreted as a good rating since the dataset does not provide information on the user rating. In reality, this isn't true because a visit could be a bad experience and the user actually dislikes the location.

The simulation is carried out in the same style that is performed by Bao et. al. The dataset is partitioned into two parts: one part is denoted as the training set and the other part is the test set. The training set is the set of user histories that the CF and MAB will train on. In the test set, each user check-in will prompt a set of recommendations. If the actual venue a user checks into is part of the recommended set of venues, it is considered that the venue is recovered.

Two metrics will determine the performance of the algorithm: the precision and the recall.

$$precision = \frac{number\ of\ recovered\ ground\ truths}{total\ number\ of\ recommendations}$$

$$recall = \frac{number\ of\ recovered\ ground\ truths}{total\ number\ of\ ground\ truths}$$

The simulation is conducted in a way to deal with the impracticalities of the offline portion of CF. In practice, this would be run on a daily basis as new check-ins would potentially result in new experts and user preference weightings. In application, this is suitable because the system performs in real-time and check-ins occur naturally. However, in the simulation, each offline calculation is expensive. Thus, the test set is for one day to reduce the time spent performing the offline batch process.

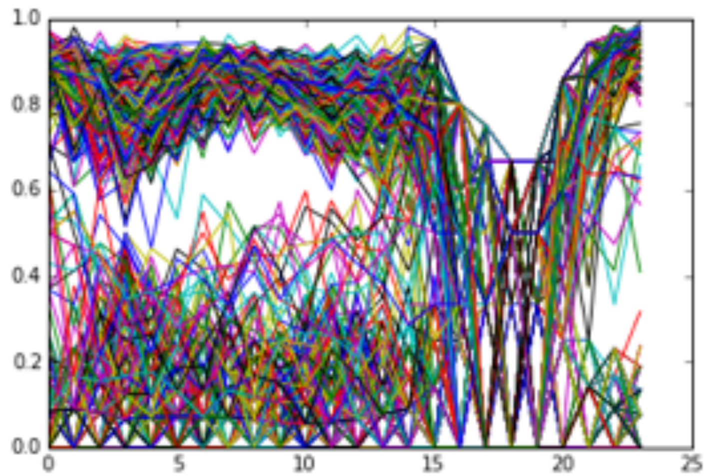
The training set covers a period from April 2012 to February 14th, 2013. The test set is for the day of February 16th, 2013. The dataset omits February 15th, 2013.

When the recommendations from the CF and MAB in series were evaluated, the results show that the performance worsened. This is probably because much of the categories only share the root of the categorical hierarchy as the lowest common ancestor and thus, their value is adjusted to zero. When the adjusted level distance is multiplied to the recommendation value, the new values become washed. Then when the system sorts the venue by their new values, all the venues that share the same value lose their positional significance. Their ordering is now dependent on how the system decides to sort elements of the same value.

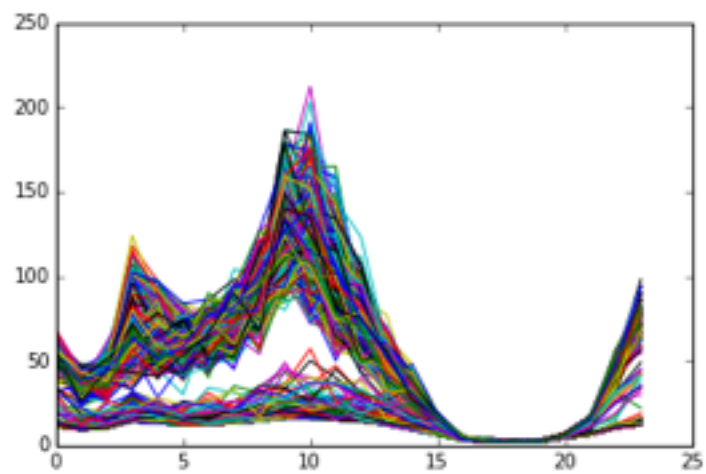
Recommendations	CF	CF+MAB
5	P = .015, R = .075	P = .0083, R = .0416
10	P = .01, R = .1	P = .005, R = .05
20	P = .002916, R = .058	P = .00625, R = .125

Two observations should be considered as explanations for the lack of significance of temporal context using the category hierarchy as a basis for rewards:

- Category popularity does change over the course of the day but all categories change similarly.
- The nature of the categories hierarchy is flat. In general, most are the same level distance apart and its scales the recommendations virtually by the same scale.



Above is the graph of the values of the arms and below is the graph of counts of all the arms throughout the hours of a day.



The hours along the x-axis are in GMT timezone and Japan is 9 hours ahead. Notice that for the counts, they all have the same relative pattern. And though there are differences, they don't differ much from one another. The values of the arms are very noisy. However, there are some arms that have low values throughout the day. These are typically unfrequented locations such as post offices and civic venues. This does not mean they are not necessarily visited at all but users probably don't see the appeal of declaring a check-in at these venues.

IV. CONCLUSION

The temporal context added by the multi-armed bandit does not show significant value to recommendations provided. There is probably some value in the temporal context but they most likely are not well represented by level differences in the category hierarchy.

I learned that constructing the rewards for a bandit is as important as the runtime attributes (O(log n) regret, etc.).

ACKNOWLEDGMENT

I would like to thank Dr. Mihaela van der Schaar and Ahmed Ibrahim for the introduction to recommendation systems and reinforced learning and the interest that it has sparked.

REFERENCES

(1) Jie Bao , Yu Zheng , Mohamed F. Mokbel, Location-based and preference-aware recommendation using sparse geo-social networking data, Proceedings of the 20th International Conference on Advances in Geographic Information Systems, November 06-09, 2012, Redondo Beach, California .

(2) P. Matikainen, P. M. Furlong, R. Sukthankar and M. Hebert "Multi-armed recommendation bandits for selecting state machine policies for robotic systems", *Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE*, pp.4545 -4551

(3) Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the nineteenth international conference on World Wide Web (WWW-10)* (pp. 661–670).

(4) J. Kleinberg. Authoritative sources in a hyperlinked environment. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998. Extended version in Journal of the ACM 46(1999). Also appears as IBM Research Report RJ 10076, May 1997.

(5) Dingqi Yang, Daqing Zhang, Vincent W. Zheng, Zhiyong Yu. Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. IEEE Trans. on Systems, Man, and Cybernetics: Systems, (TSMC), 45(1), 129-142, 2015.

(6) John Myles White. Bandit Algorithms. O'Reilly Media. December 2012.