Building A Personalized Tourist Attraction Recommender System Using Crowdsourcing

(Demonstration)

Yoram Bachrach
Microsoft Research
yobach@microsoft.com

Sofia Ceppi
Microsoft Research
soceppi@microsoft.com

Ian A. Kash
Microsoft Research
iankash@microsoft.com

Peter Key
Microsoft Research
peter.key@microsoft.com

Filip Radlinski
Microsoft Research
f lip.radlinski@microsoft.com

Ely Porat
Bar-Ilan University
porately@gmail.com

Michael Armstrong
Microsoft Research
a-mica@microsoft.com

Vijay Sharma
Microsoft Research
a-vishar@microsoft.com

ABSTRACT

We demonstrate how crowdsourcing can be used to automatically build a personalized tourist attraction recommender system, which tailors recommendations to specific individuals, so different people who use the system each get their own list of recommendations, appropriate to their own traits.

Recommender systems crucially depend on the availability of reliable and large scale data that allows predicting how a new individual is likely to rate items from the catalog of possible items to recommend. We show how to automate the process of generating this data using crowdsourcing, so that such a system can be built even when such a dataset is not initially available. We first find possible tourist attractions to recommend by scraping such information from Wikipedia. Next, we use crowdsourced workers to filter the data, then provide their opinions regarding these items. Finally, we use machine learning methods to predict how new individuals are likely to rate each attraction, and recommend the items with the highest predicted ratings.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Human Factors, Economics

Keywords

Recommender Systems, Crowdsourcing, Machine Learning

1. INTRODUCTION

Consider tourists arriving at a city that they have never visited before, and who seek information about attractions that they may find interesting. Without further information about each individual it is difficult to decide which attractions to recommend, as there is a high variance in preferences and tastes. One can recommend attractions that are generally popular, as is done by existing tourist guides and websites. Given more information about each tourist one may tailor the recommendations to individuals. For example, elderly tourists are less likely to enjoy physically demanding attractions, and introverts would probably enjoy attractions with less intense human interaction than extroverts.

Personalized recommender systems produce recommendations that are tailored to the specific individual seeking the recommendation. Thus, different people who use the system each get their own list of recommendations based on their own traits [9]. Recommender systems have been applied in various domains, such as movies, music and advertising. Typically they are built using a large dataset consisting of the traits of many individuals, a catalog of items described by some features, and the ratings that various individuals gave to various items. This dataset allows a machine learning model to correlate the traits of individuals and features of items, and thus predict how an individual user would rate a certain item. The recommender system can then recommend items that are most likely to be highly rated.

However, sometimes such datasets are not available. We show how to automate the process of building a personalized recommender system using crowdsourcing. As our target domain we have chosen the tourism domain, which has previously been explored for non-personalized recommendations [10]. In order to personalize recommendations, we must profile both attractions and users, and match each user with the attractions most suitable for her in the target location. We first find possible tourist attractions to recommend by identifying all items in Wikipedia with a specific geographic location, then ask crowdsourced workers to provide us with deep information about themselves and to express their opinions regarding the potential attractions. We then use machine learning methods to predict how new individuals are likely to rate each attraction, and recommend the items with the highest predicted ratings.

A major challenge is finding ways to incentivize workers to

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

reveal their true information and opinions about attractions. For data where the goal is to reach a consensus answer (e.g., is this attraction suitable for children?), techniques for *peer prediction* allow rewarding truthful reporting [8]. When we strive to elicit more personal beliefs (e.g., how much do *I* like this attraction) it is not clear how to extend these techniques. Our current system uses a relatively simple payment scheme, but improvements are an active area of research.

2. CROWDSOURCED RECOMMENDATION

To generate a personalized recommendation, we require a profile for the target user, consisting of *user features*. These include demographic features (age, gender, income and family status) and their psychological personality profile. The personality profile consists of the Big Five personality traits [7], a commonly accepted model in psychology for the key personality traits of an individual. These traits are measured using a short questionnaire called the Ten Item Personality Inventory [6], providing scores for each of the five traits: openness to experience, conscientiousness, extroversion, agreeableness, neuroticism. Finally, since each attraction is associated with a general attraction category (categories are: historical landmarks, architecture, art, culture, shopping, amusement parks, nature), we ask users to provide us with ratings for each of these categories. The user profile is built by asking each user to complete a *user profiling questionnaire* with 22 questions (demographic traits, personality questionnaire and general attraction category ratings). This questionnaire contains key information for deciding which recommendations are appropriate for each user. We denote the user profile features as: $U = (u_{age}, u_{gender}, \dots, u_{neuroticism})$.

Most user features, such as age or personality traits are quantitative. We partition such traits according to the user's relative location in the user population sorted by that trait. We use three bands for each trait, so the bottom third of the population in terms of a trait are denoted as *low* in that trait, the middle third as *medium*, and the top third as *high*. Given a level level $\in \{\text{low}, \text{medium}, \text{high}\}$ for a traits t of a given user i, we denote by $t_i^{\text{level}} = \text{true}$ if user i has a score of the trait t that is in the band level, and $t_i^{\text{level}} = \text{false}$ otherwise. For example, a young user i would have $age_i^{\text{low}} = \text{true}$ and $age_i^{\text{high}} = \text{false}$.

We characterize attractions by *attraction features*: the general attraction category, amount of walking required, best time to visit (daytime, nighttime, weekdays, weekends), age suitability (children, young, adult, any), popularity, and prominence (well-known, local, off-the-beaten-track). We denote the set of attraction features as A.

Identifying Attractions: We first generate a catalog of potential attractions in a target destination, each tagged with the attraction features. We derive a list of possible attractions from Wikipedia. Given a target destination (such as "London", "Rome" or "Tokyo"), we take a snapshot of Wikipedia and extract all articles that are tagged with a location in the target area using a reverse lookup in the Bing Maps API. We then use the Amazon Mechanical Turk (AMT) crowdsourcing platform to ask people to validate entries and tag them with the attraction features. For each attraction, several crowdsourced judges are paid to indicated whether this is indeed a tourist attraction, and express their opinion regarding each of the attraction features. We aggregate the opinions using majority voting. For example, if a majority of workers asked about an attraction indicate that it is a shopping attraction, we categorize it as such. This process yields a catalog of potential attractions in the target destination, which we call the *attraction catalog*.

Obtaining Ratings: We predict how a user would rate an attraction by generating a dataset D using crowdsourcing. We use AMT to source a pool of workers. We ask each worker to complete a user profiling questionnaire, and to rate a set of 20 attractions selected at random from the attraction catalog. Each such rating is stored as a triplet of the form (U, A, r), where U are the features of the rating user (captured by the user profiling questionnaire), A are the attraction features from the attraction catalog and r is the rating that this user gave to this attraction.

Generating Recommendations: After obtaining the dataset D, we apply linear regression to build a prediction model M. The model M predicts the rating that a user u', characterized by the user features, would give to an attraction a', characterized by the attraction features. To generate a personalized recommendation for a new user, we ask the user to fill in the user profiling questionnaire and use M to predict the rating that this user would give to each attraction in our catalog. We sort the attractions from the highest predicted rating to the lowers predicted rating, and recommend attractions in the top of this list.

Conclusion: We proposed a crowdsourced personalized attraction recommender system. Several questions remain open. Can other machine learning methods, such as matrix factorization, improve the recommendations? Does our methodology generalize to other domains? Can we use sketching [5, 4, 2, 1, 3] to improve space requirements and runtime?

3. REFERENCES

[1] Y. Bachrach and R. Herbrich. Fingerprinting ratings for collaborative filtering- theoretical and empirical analysis. In *SPIRE*, 2010.

[2] Y. Bachrach, R. Herbrich, and E. Porat. Sketching algorithms for approximating rank correlations in collaborative filtering systems. In *SPIRE*, 2009.

[3] Y. Bachrach and E. Porat. Sketching for big data recommender systems using fast pseudo-random fingerprints. In *ICALP*. 2013.

[4] Y. Bachrach, E. Porat, and J. S. Rosenschein. Sketching techniques for collaborative filtering. In *IJCAI*, 2009.

[5] G. Feigenblat, E. Porat, and A. Shiftan. Exponential time improvement for min-wise based algorithms. *Information and Computation*, 2011.

[6] S. D. Gosling, P. J. Rentfrow, and W. B. Swann Jr. A very brief measure of the big-five personality domains. *Research in personality*, 37(6):504–528, 2003.

[7] T. A. Judge, C. A. Higgins, C. J. Thoresen, and M. R. Barrick. The big five personality traits, general mental ability, and career success across the life span. *Personnel psychology*, 52(3):621–652, 1999.

[8] N. Miller, P. Resnick, and R. Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.

[9] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[10] H. Zhang, E. Law, R. Miller, K. Gajos, D. Parkes, and E. Horvitz. Human computation tasks with global constraints. In *CHI*, pages 217–226, 2012.