Power of Human Curation in Recommendation System

Yuchen Liu
Pinterest, Inc & UCLA
yuchen@pinterest.com

Dmitry Chechik
Pinterest, Inc
dmitry@pinterest.com

Junghoo Cho
UCLA
cho@cs.ucla.edu

ABSTRACT

This paper introduces human curation signals and demonstrates incorporating human curation signals improves the relevance of state-of-art recommendation system models by up to 30% by experiments on a large-scale Pinterest dataset.

1. INTRODUCTION

Personalization is an important task for many online services. There are two important types of signals that these services leverage: *explicit feedback* (e.g. star ratings) and *implicit feedback* (e.g watching a video). Here we introduce an important type of implicit signal, **human curation**. Nowadays more services are encouraging users to organize their interested objects: Youtube enables users to arrange their favorite videos into playlists; Spotify supports customized playlists; Pinterest users can put their Pins onto boards. We call the organization of objects by users **human curation**. The value of human curation is two-fold: 1) it facilitates a better understanding of the user's preferences, as the self-organization process sends strong explicit signals from users to show their preferences; 2) it reveals more information about objects on the service, as the co-occurrence of objects in the user collections is a strong signal of a close relationship among multiple objects. In this paper, we explore how to leverage the human curation signals for recommendations models, both to infer users' preferences and to obtain relationships among objects; and we evaluate their effectiveness on a large-scale Pinterest dataset. Experiments show the performance of the models improves by up to 30%, demonstrating the usefulness of human curation signals.

2. PROBLEM DEFINITION

We work on a standard recommendation system problem: provide personalized item recommendations for each user given observed preferences. The observed preference is in the form of a positive action on an item (as implicit feedback), e.g. saving a Pin on Pinterest. We use a binary variable r_{ui} to represent the observations: $r_{ui} = 1$ if a positive action happens on item i from user u, otherwise $r_{ui} = 0$. The goal is to predict the preference $\hat{r}_{uj}$ for user u on unseen item j, and thereafter to produce a list of item recommendations with the highest predicted preferences for each user. For human curation, a user u creates a set of his or her own collections, denoted $C(u)$. For each positive action observed from u as $r_{ui} = 1$, the item i is also curated by the user to one of his or her collections m, $m \in C(u)$. We could also use $r_{mi} = 1$ to denote this positive action observation.

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2889350.

3. MODELS

Here we introduce how we use human curation signals in two types of state-of-art collaborative filtering models in recommendation system (neighborhood and latent factor models).

3.1 Neighborhood Model

State-of-Art Model The neighborhood model is a classic collaborative filtering approach and extensively studied in the literature [2, 3]. We focus on an item-based neighborhood model, which performs better and is more efficient [4].

The critical step in the item-based neighborhood model is to compute the similarity between items i and j, denoted s_{ij}. For each item i, the model computes and keeps the set of top-K most similar items denoted $S(i; K)$. To make a prediction $\hat{r}_{u,j}$ for the preference of user u on item j, the model looks at the user u's historical preferences on items similar to item j, where typically a weighted sum scheme is applied to make a prediction:

$$\hat{r}_{u,j} = \frac{\sum_{k \in S(j;K)}(s_{j,k} * r_{u,k})}{\sum_{k \in S(j;k)}(s_{j,k})}$$

A commonly used similarity measure between two items is cosine similarity: an item i is thought of as an N-dimensional vector $\vec{i_u}$ (N is the number of users) where each dimension is a user's observed action on item r_{ui}; the similarity between two items is measured by computing cosine similarity of the two vectors as $s_{ij} = cosine(\vec{i_u}, \vec{j_u})$.

Model with Human Curation Signal Human curation signals can help better measure item similarity. Intuitively, the similarity of two items has higher confidence when a user puts them into the same collection than the case when the user simply "likes" both. The item i can be considered a vector in the collection-space instead of user-space: a M-dimensional vector $\vec{i_c}$ (M is the number of collections by all users) and each dimension is whether each collection contains this item r_{mi}. The cosine similarity can be computed as $s_{ij} = cosine(\vec{i_c}, \vec{j_c})$.

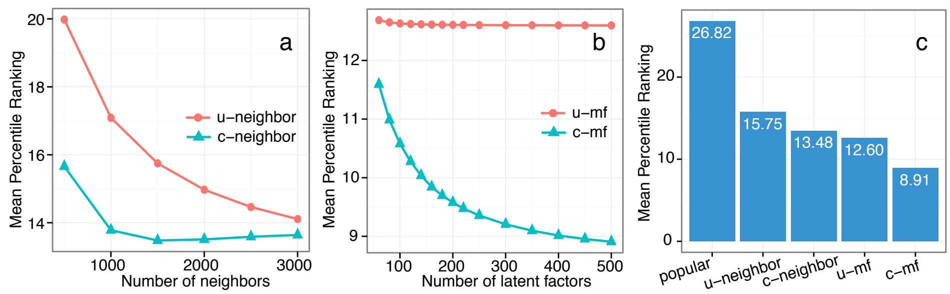

Figure 1: Experiment Performance

3.2 Latent Factor Model

State-of-Art Model Latent factor models have gained popularity due to their accuracy and scalability [4]. A latent factor model maps a user u and item i into the same K-dimensional latent factor space: a user-factors vector $x_u \in R^K$ and an item-factors vector $y_i \in R^K$. The prediction for a user-item pair is the dot product of the user-factor vector and the item-factor vector: $\hat{r}_{ui} = x_u^T y_i$.

We adopt a matrix factorization model on the implicit feedback dataset. Hu et al. [1] perform matrix factorization on the user-item preference matrix and introduce a confidence variable f_{ui} for each user-item pair r_{ui}, including all missing observations. The model optimizes for

$$\min_{x_*, y_*} \sum_{ui} f_{ui}(r_{ui} - x_u^T y_i)^2 + \lambda(\|x_u\|^2 + \|y_i\|^2)$$

where λ is the regularization parameter and f_{ui} is the confidence for each observation. A plausible choice for f_{ui} is $1 + \alpha r_{ui}$, where α is a confidence parameter.

Model with Human Curation Signal Both steps of parameter estimation and prediction can be enhanced with human curation signals. For parameter estimation, we map each collection instead of user to the latent factor space and matrix factorization is performed on the collection-item preference matrix r_{mi} instead of the user-item matrix. Each collection m is mapped to a collection-factors vector z_m. For prediction, as a user u owns multiple collections $C(u)$, the user-item preference is predicted by aggregating predictions of all collection-item pairs:

$$\hat{r}_{u,i} = \max_{m \in C(u)} \hat{r}_{m,i} = \max_{m \in C(u)} z_m^T y_i$$

4. EXPERIMENTS

Experiment Setup We collected our dataset from Pinterest where users curated their Pins (as items) onto different boards (as collections). We randomly sampled 96K users and collected 28 million Pins from a total of 856K boards they created. There are 460K unique Pins in the dataset. We performed a 4:1 random split for model training and testing. Models are implemented on Spark and the models used in the experiment are as follows:

1) Baseline model: popularity model (popular). This model doesn't provide any personalization at all as the prediction of item for any user is its popularity.

2) Neighborhood models: user based (u-neighbor) and collection based (c-neighbor) models introduced in Section 3.1.

3) Matrix factorization models: user based (u-mf) and collection based (c-mf) models introduced in Section 3.2.

Evaluation Method We adopted an evaluation methodology based on Mean Percentile Ranking (MPR) [1]. MPR is a recall-based evaluation metric that evaluates a user's satisfaction with an ordered list of recommended items, due to the lack of negative feedback in the problem setting. Specifically, each model is responsible for producing a list of recommended items for each user sorted by predicted preferences. We use $rank_{ui}$ to denote the percentile for item i within the ordered list of the user u. MPR is the average of $rank_{ui}$ for all positive actions in the test set where $r^t_{ui} = 1$.

Performance

Neighborhood Models We used different K values (the number of neighbors) ranging from 500 to 3000 and show the MPR for each model in Figure 1a. The **c-neighbor** model outperforms the **u-neighbor** model by up to 15%. As the prediction function is the same for both models, the better performance indicates that the item similarity measured in the collection-space is more accurate than in the user-space.

Matrix Factorization We used cross-validation to determine the value of α (the confidence parameter) and λ (the regularization parameter). We used different K values (the number of latent factors) ranging from 50 to 500 and shows the MPR for each model in Figure 1b. The **c-mf** model outperforms the **u-mf** model by up to 30%. They both perform better when the number of latent factors increases.

Overall Performance Figure 1c shows the performances of all 5 models. In the literature, in general the matrix factorization models perform better than the neighborhood models, while they are all better than the baseline popularity model. Models utilizing human curation signals outperform standard models in both types, by up to 30%. The matrix factorization model with human curation signals is the clear winner among all models.

5. CONCLUSION

We introduced **human curation** signals and proposed collaborative filtering models utilizing them to make personalized recommendations. Experiments on a Pinterest dataset validated the effectiveness of the proposed models and showed that the human curation signals significantly improved the overall effectiveness of recommendations by helping us better understand item relationships and user preferences.

6. REFERENCES

[1] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, 2008.

[2] P. Resnick et al. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW*, 1994.

[3] B. Sarwar et al. Item-based collaborative filtering recommendation algorithms. In *WWW*, 2001.

[4] X. Su and T. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009.