13 Discovering Community User Latent Behavior: Comparing ARM and LDA (WWW 2009)

13.1 Wen-Yen Chen, Jon-Chyuan Chu, Junyi Luan, Hongjie Bai, Yi Wang, Edward Y. Chang

13.2 Motivation

Users of social networking services can connect with each other by forming communities for online interaction, and users have great need for effective community recommendation in order to meet more users.

13.3 what

Association rule mining (ARM) is used to discover associations between sets of communities that are shared across many users. Latent Dirichlet Allocation (LDA) models user-community co-occurrences using latent aspects. Orkut data set consisting of 492,104 users and 118,002 communities. Parallelize LDA on distributed computers.

13.4 how

1. Recommender systems can be classified into two categories: content-based filtering and collaborative filtering. Content-based filtering: user profiles and descriptions of items. Collaborative filtering (CF): information about similar users' behaviors.

2. ARM can discover explicit relations between communities based on their co-occurrences across multiple users. LDA models user-community co-occurrences using latent aspects and makes recommendations based on the learned model parameters. ARM has the problem of sparseness in explicit co-occurrence. LDA has the problem that implicit co-occurrence is not always inferred correctly. This is the problem of noise in inferred implicit co-occurrence.

3. **Membership size**: would explicit relations be more effective at recommendations for active users, ones who have joined many communities. **Community size**: would implicit relations be more effective at recommend-

ing new or niche communities with few members? **Membership spread**: would explicit relations be more effective at recommendations for a diverse user, one who is involved in a miscellaneous set of communities? **Community spread**: would implicit relations be more effective at recommending umbrella communities, those composed of many smaller, tighter subcommunities or many non-interacting members.

4. LDA used in document modeling, assumes a generative probabilistic model in which documents are represented as random mixture over latent topics, where each topic is characterized by a probability distribution over words. LDA generative process consists of three steps: (1) for each document, a multinomial distribution over topics is sampled from a Dirichlet prior; (2) each word in the document is assigned a single topic according to this distribution; (3) each word is generated from a multinomial distribution specific to the topic.

5. ARM: view each user as a transaction and his joined communities as items. Employ FP-growth for mining frequent itemsets and use the discovered frequent itemsets to generate first-order association rule. With the rules, we can recommend communities to a user based on his joined communities. We weight the recommended communities by summing up each corresponding rule's confidence.

6. In LDA, user-community data in entered as a membership count where the value is 1 (join) or 0 (not join). Estimate parameters using Gibbs sampling, then infer the community recommendation from the model parameters.

7. For each occurrence, the topic assignment is sampled from:

$$P(z_i = j | w_i = c, z_{-i}, w_{-i}) \propto \frac{C^{CZ}_{cj} + \beta}{\sum_{c'} C^{CZ}_{c'j} + M\beta} \frac{C^{UZ}_{uj} + \alpha}{\sum_{j'} C^{UZ}_{uj'} + K\alpha}$$

where $z_i = j$ represents the assignment of the i-th community occurrence to topic j, $w_i = c$ represents the observation that the i-th community occurrence is the community c in the community corpus, z_{-i} represents all topic assignments not including the i-th community occurrence, and w_{-i} represents all community occurrences not including the i-th community occurrence. C^{CZ}_{cj} is the number of times community c is assigned to topic j,

not including the current instance, and C^{UZ}_{uj} is the number of times topic j is assigned to user u, not including the current instance. From these count matrices, we can estimate the topic-community distributions ϕ and user-topic distribution θ by:

$$\phi_{cj} = \frac{C^{CZ}_{cj} + \beta}{\sum_{c'} C^{CZ}_{c'j} + M\beta}$$
$$\theta_{uj} = \frac{C^{UZ}_{uj} + \alpha}{\sum_{j'} C^{UZ}_{uj'} + K\alpha}$$

where ϕ_{cj} is the probability of containing community c in topic j, and θ_{uj} is the probability of user u using topic j. The algorithm randomly assigns a topic to each community occurrence, updates the topic to each occurrence using Gibbs sampling, and then repeats the Gibbs sampling process to update topic assignments for several iterations.

8. Communities can be ranked for a given user according to the score ξ:

$$\xi_{cu} = \sum_{z} \phi_{cz}\theta_{uz}.$$

Communities with high scores but not joined by the user are good candidates for recommendation.

9. Parallelization

10. Evaluation Metric and Protocol: the metrics for evaluating recommendation algorithms can be divided into two class: (1) Prediction accuracy metrics measure the difference between the true values and the predicted values. Commonly used metrics include Mean Absolute Error

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |f_i - y_i|,$$

and Root Mean Square Error

$$RMSE(\theta_1, \theta_2) = \sqrt{\frac{\sum_{i=1}^{n}(x_{1,i} - x_{2,i})^2}{n}}.$$

(2) Ranking accuracy metrics measure the ability to produce an ordered list of items that matches how a user would have ordered the same items,

including **top-***k* recommendations and Normalized Discounted Cumulative Gain (NDCG).

11. For each user, we randomly withhold one joined community c from his original set of joined communities to form user u's training set. Second, for each user u, we select k-1 additional random communities that were not in user u's original set; the withheld community c together with these k-1 other communities form user u's evaluation set of size k. The objective is to find the relative placement of each user u's withheld community c.

12. Analysis of latent information learned from LDA: For whom LDA ranks better than ARM, the topic distributions of their joined communities tend to be more concentrated. For whom ARM ranks better, the topic distributions of their joined communities tend to be more scattered. (Plot: x: topic index, y: probability, for community, user)