**Trust in recommender system to solve cold start problem**

Weifeng Ma

SJSU CMPE 297

Recommendation systems are complex artificial intelligence systems that are designed to provide a prediction to users based on a preference. Recommendation systems require large data and time to train. Recommendation systems are heavily used in everyday life. For example, online shopping sites often recommend a list of items based on the user's history. In social networking apps, there is a feature to show those people that users might know. Likewise, when users watch videos on YouTube, the right side of the web has a bunch of recommended videos. Recommendation systems are widely useful because they save users a lot of time on the search, and they can efficiently provide services to users.

Although recommendation systems are now in heavy use, some recommendation systems provide services that are not accurate enough that recommended outcomes sometimes are not what users want. Sometimes, the systems cannot provide outcomes because of a lack of data. For example, book recommendation systems usually contain mass data because of different languages, different editions, different publishers, etc. Due to the volume of data, many book recommendation systems do not have enough samples, resulting in recommendations that do not match the real situation. The algorithms and models of recommendation systems are getting more and more advanced. The paper I read is *Survey on Recommender Systems Incorporating Trust*. What the authors want to do is to improve the performance of recommendation systems.

To find similar items, system often use matrix factorization. Matrix factorization is a collaborative filtering algorithm to find the relationship between items and users’ entities. In matrix factorization, the main data need to be divided into two parts: items and users. The system would use these two data to compute dot products. The result indicates a rating of a user for an item. After all the computation is done, the system would have a table that contains all the key information about users. Likewise, the system can use the matrix factorization table to find similarity between items. To make that approach, people often did it by using similarity algorithms such as cosine similarity, K-nearest neighborhood (KNN). By finding the similarity, the system can perform similar item recommendations. However, this approach often leads to a problem, cold start.

One of the major problems of recommendation systems is the so-called cold start problem. Whenever a new user or items starts a recommender system, the system cannot provide recommendations because data is not available and similarity calculation could not be performed. The simplest way is to solve this problem is to let the recommendation system "guess" the user's preferences without any user action. The system automatically generates categorized top results and mines various lists as candidate sets offline, and then customizes a general recommendation candidate set by considering the novelty, diversity, freshness, and other dimensions of the recommendation results. Finally, we capture the user's click feedback behavior data in real-time and iterate continuously to gradually generate results that meet the user's personalized taste. This method is simple to implement but requires high timeliness for offline systems.

In the research paper Survey on Recommender Systems Incorporating Trust, the authors use trust-aware to solve cold starts. Trust is defined as the mutual social relationship that two users have with each other. Impersonal aspects of trust are considered such as integrity, benevolence, predictability, and competence. Trust helps to mollify the cold start problem of the recommender system and enhance the coverage of the recommender system by suggesting items to the new users. Trust includes user behavior data. User behavior data included the number of clicks on a particular website, navigation patterns through the website, and duration of visits to particular websites. After the data on trust is collected, we can reconstruct the trust-aware network by detaching trust links between several users having high correlation coefficients below a specific threshold value. We can use a latent variable called "strength level" to indicate the amount of trust that exists between two users. Integration of trust in recommender systems can have the following advantages:

• Good coverage: better coverage of similar users is one of the biggest advantages of using trust-aware recommender systems as compared to usual collaborative filtering.

• Better Accuracy: More accurate predictions for items as the trust of opinions from other users is used as a criterion.

• More Stable: These systems are more scalable than the normal recommender systems.

Graphical user interface

Description automatically generated with low confidence

Figure 1. Beis Sturctre of Trust aware recoomender system (TARS)

In the TARS, there are two inputs: the trust matrix of n users and the user item rating matrix. Trust matrix can be found by explicitly asking the user about their preference. This could be done by using general surveys, and question answers. The rating matrix is used to find the nearest similar neighbors of the users to whom we want to recommend items. The output is a combination of the rating matrix of these users.

Recommender systems are becoming more and more personalized across many of the world's largest platforms and applications. We might not notice, but we use recommender systems on daily basis. The paper introduces how to deal with cold start problems. Recommendations for the new user can be given by trust-based recommender systems. Trust-aware recommender systems are used to incorporate trust in recommender systems so that shilling attacks couldn’t bias the suggestions of the recommender systems. We can also use trust-aware to remove the cold start problem so that new users could get suggestions from the users they trust in the system.