Trust in Recommender Systems

Isaac Yoakum
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
Yoaku005@D.umn.edu

ABSTRACT

Providing meaningful and accurate suggestions increases both user satisfaction and business revenue. However, it is especially challenging to provide these type of recommendations to new users of any particular system. In this paper, I describe research that aims to optimize predictions of how a user will rate an item. Systems that can make good guesses about how a user will rate an item are better able to provide meaningful recommendations. T he research I describe used Ant Colony Optimization and trust aware collaborative filtering, and the results from comparing this technique to other approaches on three different systems yielding the best results for cold-start users.

Keywords

Trust, Recommender System, Collaborative filtering

1. INTRODUCTION

In the 21st century, it is very uncommon to go out to eat, go on a vacation, or go see a movie without being recommended a place or reading reviews online. These online reviews are helpful because you want to find the right location for your outing that is best fit for you. This is a helpful example to show what recommender systems are all about.

A recommender system is a system that recommends a user something based on what they are doing or looking for. A specific example could be found on the service Netlix. When searching through the movies, you will come across a section that is called recommended movies. These movies are generated by a series of background algorithms that take many factors into account while suggesting movies that the recommender system may find to be something a specific user would like to watch. Although it may seem to just be another category in a movie website that suggests you new movies, it is actually a very important feature that can actually determine the worth of a movie viewing application.

To show the importance of recommender systems, Netflix held a competition in 2009 that was open to the public that offered one million dollars to the individual or team that created the best collaborative filtering recommender system. The team that won ended up besting Netflix's algorithm by ten percent to predicting ratings for movies. This example shows how willing companies are to have the best-of-the-best recommendation systems because it will make them significantly more money in the future.

Now that we understand the relevance of the systems, we need to understand the background of what recommender systems are. In this paper, I will be describing research about one type of recommender system. This recommender system uses various techniques to produce the most accurate rating predictions for a service. This paper will introduce two helpful techniques, adding trust and meta-heuristic techniques to a recommender system for improving predictions of user ratings. Once we understand the recommender system and the process of predictions, we will look at some data sets and compare this technique to other state of the art recommendation systems to check the quality of the predictions.

UMM CSci Senior Seminar Conference, April 2018 Morris, MN.

2. BACKGROUND

To understand the recommendation paper that we will talk about in detail, there are three ideas we need to understand.The first thing we will talk about is the type of recommendation system we use. The second thing we cover is what is trust in a recommendation system. Thirdly we will introduce and talk about meta-heuristic algorithms and their benefit to recommendation process.

2.1 Filtering Techniques

There are many different type of recommender systems. Some of the most popular are known to be collaborative filtering, content based filtering, hybrid filtering, demographic filtering, and knowledge filtering. These prove to be beneficial in their own ways, but the filtering we will look at is collaborative filtering. Collaborative filtering has proven itself to be the most useful filtering system in comparison to others. Collaborative filtering has two different types of techniques. The first technique is the user-to-user collaborative filtering.The user-to-user technique is the recommendation process that looks at two different users to give the most accurate item recommendation to the user. It compares the users by similarities and if they have the most similarities, user one will be suggested something user two likes using that collaborative filtering process. Item-to-item is the second collaborative filtering technique. The recommendation process is similar, but doesn't need two users. They take a target user for a recommendation,look at what items they liked in the data set, and then recommend a similar item to the liked item.[4] Both of these techniques are very accurate in recommendation systems, but the paper I discuss here

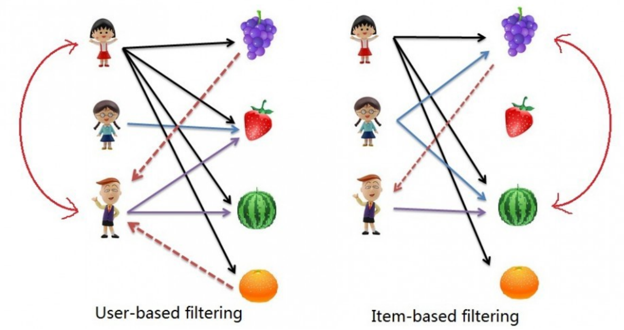


Figure 1: item-item vs user-user.

will be using the user-to-user technique.

2.2 Trust in Recommender Systems

The next important piece to understand about recommendation systems is the trust factor that can be added as additional information in the process. Trust is known to be a hard thing to define.In the context of recommender systems, one way you can look at it is being context specific interpersonal trust, which is where a user must trust another user with respect to a specific situation [5].

Trust can be beneficial in recommendation processes in many ways including providing rich side information, eliminating malicious attacks, help with sparsity problems, and strengthening recommendations for cold-start users. As these can be difficult to understand, I will provide definitions of each benefit. Rich side information is an important feature because it adds more information to the recommendation system. It is always beneficial to a system to have more information on a user to create more personalized information. Malicious attacks can be defined as accounts being made for the core reason of providing bad reviews.[1] An example could be a company releasing a new item, and the company's competition creating accounts to release thousands of bad reviews to make the item look worse than it possibly is. On the contrary though, the company releasing the item could also create thousands of good reviews to make their item look good. Any type of reviews being altered for the purpose of business is considered a malicious attack and trust will help eliminate this. The sparsity issue is a problem where items are not heavily reviewed and this can create havoc when trying to have the item recommended to people with the core data as just reviews. Cold-start is an issue where users are new and have given little to no reviews. This causes problems because without trust, you cannot assume much of the user and it's hard to know if the user is legitimate and therefore can't make accurate predictions for the user. After understanding the benefits of trust, now we need to understand the types of trust that can be made for users and items.

2.2.1 *Types of Trust*

There are two types of trust made in these recommender systems. The first type of trust is known as explicit trust. Explicit trust is trust that is provided by the user and it is then known to the system. Explicit trust is the best type of trust because it is trust that there is no doubt about because the user declares it. The second type of trust in these systems is implicit trust. Implicit trust is different than explicit trust because it is implied. The implication of trust is made by using information known about the user such as data in their profile or the types of reviews they have made.[5] A helpful example of the differences of trust can be used while examining the music application Spotify. If you want to look at ways trust can be gathered by Spotify, you can see the difference of explicit and implicit in their design. When creating an account in Spotify, you first are given the option to select some genres you like, some artists you enjoy listening to, and some hit songs you like. This information gathered onto your account can be used to create implicit trust between users by comparing you to other users that have the same type of data on their account. An explicit example using Spotify would be how you can create friendships in Spotify. Many people use it as as social network and can add and follow people based on being friends in real life or people that have similar music. This data can be used and it is explicit because they are declaring users they know or like. Understanding the way trust is gathered is important because there can be many different ways, but for this paper we use the implicit and explicit methods.

2.3 Meta-heuristic Algorithm

The last term to understand before covering the researched paper is meta-heuristic algorithms. In computer science and mathematical optimization, a meta-heuristic is a higher-level procedure or heuristic designed to find, generate, or select a heuristic (partial search algorithm) that may provide a sufficiently good solution to an optimization problem, especially with incomplete or imperfect information or limited computation capacity. Metaheuristics sample a set of solutions which is too large to be completely sampled. Metaheuristics may make few assumptions about the optimization problem being solved, and so they may be usable for a variety of problems.[3]

2.3.1 *Ant Colony Optimization*

Now that the definition of Metaheuristic is known, we will introduce the metaheuristic algorithm the paper covers which is ant colony optimization also known as ACO. ACO was originally introduced in 1992 by Marco Dorigo. It was first used as a metaheuristic method to optimize the best path on a graph. This behavior is exploited in artificial ant colonies for the search of approximate solutions to discrete optimization problems, to continuous optimization problems, and to important problems in telecommunications, such as routing and load balancing. With the optimization goals of ACO, it was found to be efficient and used in the future for many different optimizations such as in 2013 Liu et al. proposed an ACO method that mimics ants and studies behavior of people searching for useful information in the web resembles ants. Also in 2012 Mocholi, Martinez, Jaen, and Catala proposed a generic semantic multi-criteria ant colony algorithm to generate music playlists using ontologies. Ant colony optimization has been found to be a useful when looking to optimize a solution space in a problem which is what the paper covers. In this paper we will look at ACO in advanced recommender system that uses trust and collaborative filtering to address some of the problems in regular recommendation systems.

3. ANT COLONY OPTIMIZATION PAPER

There are many advanced recommender systems being introduced to the world due to the demand of bettering online business. The recommendation system paper I will go over in detail is titled "Trust Aware Collaborative Filtering Recommender System based on Ant Colony Optimization" and was published September 2018 by Hashem Parvin, Parhem Moradi, and Shahrokh Esmaeili. This recommender system uses trust statements as rich side information with ant colony optimization[2]. What this suggests is that by using trust statements along with already existing rating values of the user and target users in the database, a more accurate recommendation can be made with the search method of ACO while simultaneously eliminating problems in collaborative filtering methods such as cold-start users and sparsity. To understand the paper in detail I will be dividing the paper up into two key parts. First I will give a deep understanding of ant colonies and how they represent the algorithm, I will then go over the recommendation process and cover each step in detail starting from the inputs and ending with the prediction phase.

3.1 Understand Ant Colonies

To understand the meta-heuristic ant colony optimization algorithm, it is important to understand what it is mirrored after. By just analyzing the name, you can already predict it is based off of an ant colony, but furthermore, it is based off of an ant colony's technique to locate and find the most efficient path to food in a given location. So by this description, it really a meta-heuristic technique to find an optimized path to a target value or point. A meta-heuristic algorithm is an algorithm that aims to find a possible solution for complex and optimization tasks. It searches through a set of possible solutions to find the optimal region of the search space and discovers a near optimal solution in reasonable time. [2]

Now that we understand meta-herusitics, we can dive deeper into the ant colonies. The technique starts with all of the ants in the colony leaving their nest/home to go search for food [2]. When an ant takes a route to go search, it leaves a trail behind it called pheromones. Pheromones are a chemical that ants leave on the ground when walking around. Pheromones eventually evaporate after a certain amount of time, which can be used as an indicator of how efficient the route is. If an ant finds an efficient route to a food source, it will then return to and from the nest and food source more frequently than other ants, which then leaves a stronger scent of pheromones on the ground. This scent is eventually analyzed by other ants and the fastest trail is now followed. This trail that is followed is now used more by ants until there is a dominant path that smells strongly of pheromones (see Figure 2). This is now the most efficient path to and from the food source and it is followed by the ants to collect resources for the colony.

The important piece in this technique is the evaporation of the pheromones. If the pheromone did not leave the ground, then it would be impossible to decipher the fastest route and thus ants would just follow any given route without increasing efficiency. In summary, the ant colony optimization algorithm follows ant colonies technique very closely to generate an optimized trial and error optimization.

3.2 Process of the Filtering

Now that we have a detailed description of ant colonies and their searching technique, we are ready to dive into the recommendation process. The recommendation process (shown in Figure 3) is going to be divided into four important steps being listed below.

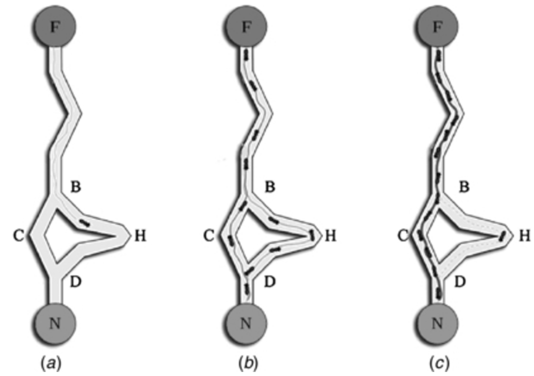


Figure 2: Ant Colony Logic.

- Inputs
- Ranking process
- Weighting process
- Prediction process

The first step is declaring the inputs to the recommendation process. When we declare the inputs, we then look at the ranking process, which looks to filter out the dissimilar users and send the now ranked users to the weighting process. The weighting process applies the ant colony optimization technique to filter through the ranked users to find the optimized list of users through ants search technique. The last step is analyzing the weighting results and creating a prediction using an equation that takes results of weighting process into consideration.

3.2.1 Inputs

The first step to the recommendation process is looking at the inputs. There are three inputs that go into this recommendation process. The first is the target user. The target user is important because all steps are used by comparing the target user. The next input is the rating matrix. A rating matrix can be compared to a spreadsheet that has all of the information of the target user and other users in the database. The last input to this step is the trust network. The trust network is the level of trust the target user has for users in the data set which can be used in the process to find dissimilar users or remove users with no trust from the rating matrix comparisons.

3.2.2 Ranking Process

The next step is using the inputs and starting the ranking process. To start the process, first an explicit trust network is made between the target user and the other users in the database. Once the explicit trust is added to the trust network, then implicit trust statements are added as well. Implicit trust statements are added to the trust network by looking at the available ratings of the users and target users in the rating matrix, and then using the Pearson correlation coefficient on them to declare levels of trust between users.

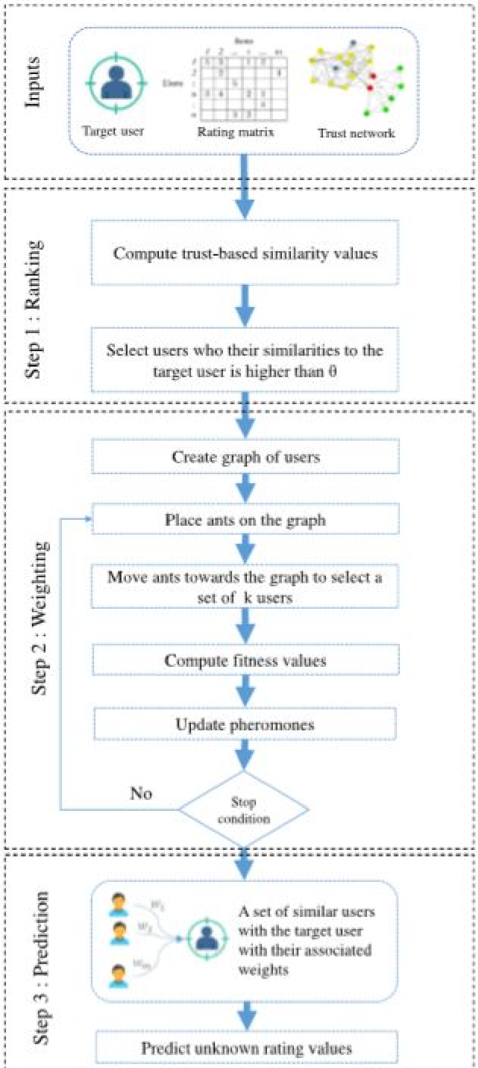


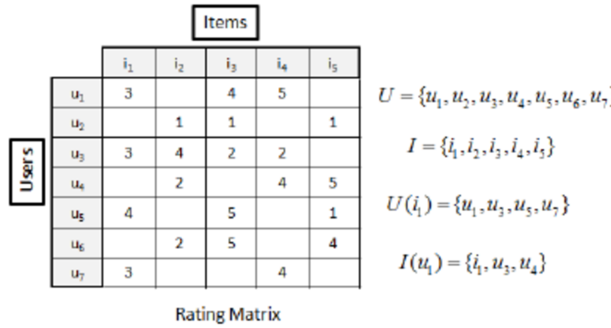
Figure 3: Recommendation process in ACO paper.

In this trust network, the nodes represent the users in the database, and the weights are representing the similarity values between the users. To then be ranked, the similar users with similarity values higher than the threshold are selected to be neighbors of the target user. The neighbors are then selected and used in the weighting process.

3.2.3 Weighting Process

The weighting process is now the next step in the recommendation system. The process involves three different steps that we will cover in detail. The first step is creating the graph of the users that will have ant colony optimization applied to it. The second will look at the placement and the movement of the ants. The third step is the updating of pheromones through iterations.

1. We begin by creating the graph. To apply the ant colony optimization, you need to create a sufficient graph that the metaheuristic algorithm can be applied to. This is done by taking a set of top similar users to the target user from the ranking process and placing



	i_1	i_2	i_3	i_4	i_5
u_1	3		4	5	
u_2		1	1		1
u_3	3	4	2	2	
u_4		2		4	5
u_5	4		5		1
u_6		2	5		4
u_7	3			4	

$$U = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7\}$$

$$I = \{i_1, i_2, i_3, i_4, i_5\}$$

$$U(i_1) = \{u_1, u_3, u_5, u_7\}$$

$$I(u_1) = \{i_1, i_3, i_4\}$$

Figure 4: Rating Matrix Example.

them into an undirected weighted graph. In the undirected graph, each user is represented by a node on the graph. With the nodes being created, then you need to apply weights between each node. The weights are are going to be representative to the similarities between the associated users in the graph. When the nodes and weights are created, the environment is now ready to have the ant colony optimization applied to it.

2. For the next step, we take the graph that was made previously, and apply ants to it. Ants are placed on the graph randomly throughout the nodes. The number of ants are calculated using the initial amount of users in the database. In the initial set up of the graph with the ants, each node is given a random level of pheromone to start the ACO process. Once the ants are initialized and the nodes are populated with pheromone levels, the ants throughout the graph with three goals in mind. The first goal is to make sure every node is touched. This is important to eliminate the possibilities of a local optima. A local optimum is a set of users that could be ranked as most similar without actually being most similar. This happens when ants get trapped in only part of the graph and therefore don't actually know if these users are the most similar users because they don't know of all of the other users in the graph. The second goal is to update each weight with pheromones. This is important because throughout the tours of the ants, they are attracted to the weights with more pheromones on them and it will help produce the accurate path to the most similar users.

3. The next step is updating the pheromone. The pheromone levels get updated as ants travel through the graph, if the values of pheromones are high, this is an indicator of a greater solution. This process is done in iterations with the pheromone evaporation strategy being put into place. Pheromone evaporation is important to understand in this step because as pheromones are updated throughout the iterations, levels will decrease the more iterations done as the connection of bad solutions decline. When the first round of ants have done their movements, the pheromones are set to the level they ended on, and the process repeats itself until having the most accurate pheromone paths. Once the algorithm has gone through its given amount of iterations, it is then concluded and the pheromone levels

on the graph are used to make a prediction.

3.2.4 Prediction process

The last step is the prediction process. This is where we look at all of the results from the ranking and weighting process and use it to create an accurate prediction for the target user. The way this is done is by looking at the results from the final iteration made in the weighting process. What is taken from the results is the levels of the pheromones. Each node is going to have a different level of pheromone and this is is the indicator of the similarity to the target user. For the actual prediction then, we take the top K users with the highest level of pheromones, and use them in a collaborative filtering user to user recommendation process. The process is done by using this equation.

$$\hat{r}_{u,i} = \frac{\sum v \in_U W_v R_{v,i}}{\sum v \in_U W_v}$$

In this equation $\hat{r}_{u,i}$ represents the target item prediction. The top part of the fraction ($\sum v \in_U W_v R_{v,i}$) is taking the sum of the pheromone levels W_v multiplied by the ratings of top users $R_{v,i}$. The bottom part of the fraction ($\sum v \in_U W_v$) is the sum of all of the pheromone levels W_v. Once we have the output number, the prediction process is now done and that concludes TCFACO process.

3.3 Tests and Results

To show the effectiveness of this recommendation technique, Parven et al. ran tests on several different real-world data sets to look at improvement of accuracy from the technique.The data sets looked at have real world users with real ratings on items. These ratings are used in comparison to prediction ratings to show the accuracy of the recommender systems tested.The data that was collected was gathered from www.Epinions.com which is a website that opinions are expressed on items; FilmTrust, which is a website that recommends movies to its users; and Ciao is data collected by Ciao research group. Each of the data sets are split into six different categories to show the effectiveness of the algorithm on different aspects of data. The six categories are cold-start users, heavy raters, opinionated users, niche items, controversial items, and all users. Cold-start users are users with fewer than 5 ratings. Heavy raters are a set of users that have rated more than 10 items. Opinionated users are users with more than 4 ratings and the standard deviation is higher than 1.5. Niche items are items with fewer than 5 ratings. Controversial items are items with a standard deviation of 1.5 or higher. All users is a set of all of the users in the data set. The tests used two types of measurements to check accuracy. The first being Mean Absolute Error which will be reffered to as MAE, and Root Mean Square Error which will be refereed to as RMSE.

Parven then tests the TCFACO against other grouped recommender systems. The three groups he tested TCFACO against are meta-heuristic-based methods which is what TCFACO is, social network-based recommenders, and rating-based methods.

The first result of comparing against meta-heuristic-based methods show that there is remarkable difference between TCFACO methods compared to the other meta-heuristic-based methods. Results also suggest that with higher number of users selected for the prediction process, the more apparent that TCFACO stands above the other recommender

	All Users	Cold start Users
Film Trust	**14.50%**	**13.60%**
Epinions	**5.30%**	**3.10%**
Ciao	0.40%	**3.90%**

Figure 5: TCFACO VS. Meta-heuristic Algorithms.

	All Users	Cold start Users
Film Trust	12.26%	**7.78%**
Epinions	**6.66%**	**7.56%**
Ciao	9.02%	**23.98%**

Figure 6: TCFACO VS. Trust based Algorithms

systems. In figure 5 you can see the percent better TCFACO was when compared to the meta-heuristics in each data set where bold represents TCFACO being the best.

The next recommender system tested against was network-based recommenders. These recommender systems are trust based and showed TCFACO had lower MAE and RMSE values compared to the other systems. This data supports the idea that by using weights of neighbor users in the prediction process results in a much higher improvement in the results. In figure 6 you can see the differences in percent of which TCFACO was better than trust aware recommenders with bold being where TCFACO was best.

The last comparison made is TCFACO to rating based methods. This data shows that in all cases of the testing, the TCFACO method achieved lower MAE and RMSE values compared to the other methods. Compared to the other methods, TCFACO improved the accuracy of results by about 12.4 percent. It is also apparent that TCFACO covers a greater range of users due to rating-based methods only being able to look at rating values compared to TCFACO which can look at implicit trust.

The important information that was gathered from all of the tests made against other recommender systems were that when compared against TCFACO, every recommendation algorithm fell short to the predictions TCFACO made for cold-start users. With cold-start being a large fundamental problem in collaborative filtering recommender systems, this is a statistic that really shows the impressiveness of the introduced recommender system.

4. CONCLUSION

Recommender systems have proven to not only be helpful to many services, but it has become necessary to compete seriously in the world of business. In this paper we covered background information of recommender systems. We explained the idea of trust to recommender systems and looked at the benefits. Lastly we went over in detail a trust-aware meta-heuristic recommender system and looked at its increase in accuracy of predicting ratings.

With intensive research, there has been a couple conclusions drawn from using an advanced recommender system such as the Trust Aware Collaborative Filtering System Using Ant Colony Optimization papers technique. The first thing to conclude is that adding trust to a collaborative filtering method can increase the accuracy in prediction and widen the pool of users to create predictions for. The second conclusion would be that using Ant Colony Optimization on a recommender system is beneficial in many specific areas of recommender systems, but most importantly in cold-start users. We found that while testing TCFACO against any other state of the art recommender system, cold-start user predictions were always more accurate. The last conclusion is that the more information given to a recommender system, the more accurate the future of predictions in recommender systems can be.

5. REFERENCES

[1] P. Moradi and S. Ahmadian. A reliability-based recommendation method to improve trust-aware recommender systems. *Expert Systems with Applications*, 42(21):7386 – 7398, 2015.

[2] H. Parvin, P. Moradi, and S. Esmaeili. Tcfaco: Trust-aware collaborative filtering method based on ant colony optimization. *Expert Systems with Applications*, 118:152 – 168, 2019.

[3] Wikipedia contributors. Metaheuristic — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Metaheuristicoldid=924483421, 2019. [Online; accessed 20-November-2019].

[4] Wikipedia contributors. Recommender system — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Recommender$_{s}ystemoldid = 922267294, 2019.[Online; accessed 13 - November - 2019]$.

[5] S. Yadav, V. Kumar, S. Sinha, and S. Nagpal. Trust aware recommender system using swarm intelligence. *Journal of Computational Science*, 28:180 – 192, 2018.