

Reciprocal Recommender Systems in Online Dating Systems

Dailon Dolojan¹

¹Department of Computer Science, University of California Santa Cruz, Santa Cruz

March 17, 2018

Abstract

A typical recommender system implements the concept of a user being recommended various items based off a user's interests. This system is commonly seen within online shopping which implements user-item recommendations to match items based off a user's interests. Although online dating sites utilizes the same process of recommending possible romantic partners similar to a typical recommender system, a successful match requires that the possible romantic partner reciprocate interest to the user. This concept of reciprocity of interest further distinguishes online dating sites as a reciprocal recommender system. Within this article, we will examine two reciprocal recommender systems and examine their approach to successfully matching online dating users with prospective relationship matches.

1 Introduction: What is a Reciprocal Recommender System?

Within the online dating industry, users create dating profiles in order to find potential romantic partners. Online dating sites often try to quantify “human attractiveness” between a user profile and other site users through the usage of boolean expressions. These expressions are generated from user inputted data based on biographical information, and responses prompted from a dating site’s questionnaire. The dating site utilizes the data from the user’s profile to create suggestions of compatible partners through the site’s matching algorithm.

A recommender system is often defined as a system which provides a user with a generated set of items that the system perceives the user will like based on inputted data. Although the traditional definition of a recommender system works for recommending movies, books, or any inanimate item a user may want to buy online, the online dating industry contains another dimension of complexity such that the potential romantic partner who is paired with the user must also like the user in order to be considered a successful match. In traditional recommender systems such as online shopping, an individual can be suggested various shopping items without the pretense of these items needing to “like” you back. This classifies online dating sites as a form of a reciprocal recommender system where not only the user has to like the recommended partner, but the recommended partner must also like the user to be classified as match.

Within this article, we will examine various forms of recommender systems for online dating sites and compare the effectiveness of each system based off successful matches.

1.1 Recommender System’s Online Dating Algorithms

In analyzing reciprocal recommender systems, it is important to differentiate between two forms of algorithms that pair user and target profiles: content-based algorithms and collaborative filtering algorithms.

1.1.1 Content-Based Filtering Algorithms

Content-based algorithms are algorithms that pair a user and potential partner by analyzing the data collected from user profiles and checking for similarities. The algorithm is based off the idea that if two users have similar content in their profiles, both users would be a successful match.

1.1.2 Collaborative Filtering Algorithms

Conventional collaborative filtering algorithms used in recommender systems often recommend objects to users based on the idea that users with similar behavior will like the same objects. This form of filtering algorithm is often seen within e-commerce, but within the context of a reciprocal system collaborative filtering is implemented to recommend a user all other profiles whose recommendations also contain A.

2 RECON Recommender System 2010¹

Within “RECON: A Reciprocal Recommender for Online Dating” (Luiz Pizzato et al. 2010), the authors of the article create a reciprocal recommender system that analyzes the interaction of a user’s *explicit* and *intrinsic* profiles of preferences within their ideal partner through content-based filtering algorithms [2]. An *explicit* profile is created by the user where the individual provides information about themselves and their preferences explicitly. In comparison, an *intrinsic* user profile is a profile of the user developed by the online dating site that collects information based on the user’s behavior interacting with the recommender site.

In using a traditional online dating site, a user creates a profile filled with inputted biographical information about the individual as well as their perceived ideal partner. The site prompts the user to provide information such as age, gender, location, the type of relationship sought after, and other various attributes. The system will then recommend potential matches to the user where the user can then browse any of the profiles to determine if the person appears to be a viable match. The user can then send a pre-defined message asking if the individual is interested and exchange contact information.

As the user interacts with a multitude of profiles, the system will develop an *intrinsic* history of his behavior on the site by examining the profiles the user decides to message. The system then uses the data collected from this *intrinsic* profile to develop a preference model for the user. This preference model is composed of all the attributes found in a potential partner that a user decides to message and the attributes associated with the potential partners that received a positive reply from the user when the user is messaged first. The preferences generated by the site is then used to calculate compatibility with a potential partner. The compatibility scores of a potential partner is then compared amongst other potential matches to output a list of recommendation for the user.

2.1 User preference’s

The online dating system creates user profiles with two key components that consist of free text information and a pre-defined list A of attributes [2]. The pre-defined list of attributes represent numeric and nominal attributes which are discretised into various attribute groups. An example of this can be seen with a user’s age being discretised into age groups such as 22-24, 24-26, and so forth.

A user x will have a profile U_x that will contain values v of a specific attribute a which is represented as

$$U_x = \{v_a : \text{for all attributes } a \in A\} \quad (1)$$

The system then generates a list M_x composed of potential partners that user x has sent a message to:

$$M_x = \{m : m \text{ is a potential partner messaged by } x\} \quad (2)$$

Preference p of a user x with values of an attribute a where n is the number of times v has appeared in M_x can then be represented by the following expression:

$$p_{x,a} = \{(v, n) : \text{for all unique discrete values } v \text{ of } a\} \quad (3)$$

¹Equations found in this section were obtained from [2]

User x 's complete set P_x of preferences can be expressed as a list of $p_{x,a}$ for every attribute A such that

$$P_x = \{p_{x,a} : \text{for all } a \in A\} \quad (4)$$

2.2 RECON's Algorithm

Once the preferences P_x of a user x is determined, RECON implements two algorithms to generate a list of recommendations. The system uses a compatibility algorithm to determine a numerical score of how well the user and the potential partner are suited for each other. After generating a list of compatibility scores with various partners, the system will implement a reciprocal recommendation algorithm. This algorithm creates a reciprocal score that reflects how well the user and the potential partner match according to their preferences and attributes.

2.2.1 Compatibility Algorithm

The compatibility score of a user and a potential partner is calculated using the following algorithm.

Algorithm 1: *Compatibility*(P_x, y)

Input: Preferences P_x , user y
Output: Compatibility score s

```

begin
  /* Return a small non-zero value if  $x$  has no preference
  (e.g.  $x$  did not send any message) */
  if  $P_x = \emptyset$  then
    return 0.001
  else
    Extract user  $y$ 's profile  $U_y$ 
    foreach Attribute  $a \in A$  list of attributes do
      Obtain the value  $v_a$  of attribute  $a$  in  $U_y$ 
      Get  $p_{x,a}$  in  $P_x$  Find  $(v, n) : (v, n) \in p_{x,a}, v = v_a$ 
      if  $n = 0$  then
        /* e.g. male user  $y$  for a strict
        heterosexual male user  $x$  */
        return 0
      else
        /*  $s_a$  is the percentage of the distribution
        that is equal to  $v$  */
         $s_a \leftarrow \frac{n}{\sum n \in p_{x,a}}$ 
    return  $\frac{\sum s_a}{|A|}$  // Arithmetic mean of all  $s_a$ 
end

```

Figure 1: RECON's algorithm to calculate Compatibility given a user x 's preferences P_x and a user y [2].

This algorithm takes user x 's preferences P_x and a potential partner user y as inputs to determine a compatibility score s . The algorithm checks to see if user x has no preferences such that $P_x = \emptyset$. If $P_x = \emptyset$, the compatibility score will result in 0.001. Otherwise, the algorithm will extract user y 's profile U_y and obtain the values v of every attribute a within the list A of attributes.

If the number of times n the value v occurred within user x 's messaging list M_x is equal to zero, then the output will be equal to zero. This illustrates that if an attribute a does not appear within user x 's preferences P_x , the attribute will return an output of zero. If $n > 0$ for an attribute a , the algorithm will calculate a compatibility score s_a for that specific attribute by dividing n from the overall sum of values that have appeared in M_x . For all $a \in A$, the algorithm finally calculates the arithmetic mean of all compatibility scores s_a to determine overall compatibility such that the sum of s_a is divided by the number A of attributes.

2.2.2 Reciprocal Recommendation Algorithm

Once the compatibility score is obtained for a user y , RECON implements a reciprocal recommendation algorithm to calculate a score that illustrates how well the preferences of a user match with a potential partner's attributes and vice versa. The algorithm intakes a number N of recommendations for user x and outputs a list of recommendations R with a list S of reciprocal scores.

Algorithm 2: ReciprocalRecommender(x, N)

Input: User x , N number of recommendations to provide
Output: List of Recommendations R , List of Scores S

begin
 Find P_x
 $R \leftarrow M_x^c$ // all users not messaged by x
 $S \leftarrow \{s_y : \forall y \in R\}$
 foreach $y \in R$ **do**
 $s_y \leftarrow \text{Compat.}(P_x, y)$
 if $s_y > 0$ **then**
 Find P_y
 /* Calculate the reciprocal score as the
 harmonic mean of the two reciprocal
 compatibility values */
 $s_y \leftarrow \frac{2}{((s_y)^{-1} + (\text{Compat.}(P_y, x))^{-1})}$
 /* sort by reciprocal score */
 $R \leftarrow \{y_1, y_2, \dots, y_j : s_{y_i} \geq s_{y_{i+1}} \forall i\}$
 if $\exists N$ **then**
 $R \leftarrow R \setminus \{y_k : k > N\}$
 return (R, W)
end

Figure 2: RECON's algorithm to generate a list of recommendations [2].

The algorithm first determines user x 's preferences P_x , the set R of recommendations that are comprised of the set M_x^c of potential partners that have not been messaged by user x , and the set S of reciprocal scores. For every potential partner $y \in R$, the algorithm calculates the compatibility score then calculates the reciprocal score as "the harmonic mean of the two reciprocal compatibility values" [2]. The algorithm then sorts the potential partners based on the highest reciprocal scores and returns the sorted list R .

2.3 Analysis of RECON

Although RECON’s recommendation algorithm has a linear performance, implementing RECON within a database of all users would result in a $O(N^2)$ time complexity [2]. Therefore to accurately test the RECON system, the team chose to use a smaller set of users. The set was comprised of users who are “of the dominant gender, who are within one standard deviation of mean age, and who live one standard deviation from the mean location” (Pizzato et al).

The two main goals set out by RECON were to understand “the role of reciprocity in the effectiveness of the algorithm” (Pizzato et al) and “how well RECON performs when it is restricted in the number of recommendations it can offer” (Ibid.).

In order to test the efficiency of the RECON recommender system, the users’ message history was collected throughout a six week time period. The first four weeks were used as training data to allow for the content based recommender system to build an *intrinsic* profile of the user and determine matches whereas the last two weeks were used to test the RECON system. Within the training set of data, 90,000 users sent approximately 1.4 million messages.

RECON’s success was then measured by the *success rate* of *known* interactions pr that have occurred with the list of sorted recommendations throughout the test period. The success rate of user x ’s list of recommendations R is the “proportion of the *known* interactions $[kr]$ (users messaged by x) that were positive (users who replied to x positively”.

$$Success(x, R) = \frac{|\{pr : pr \in R, pr \text{ replied positively to } x\}|}{|\{kr : kr \in R, kr \in M_z\}|} \quad (5)$$

The experiment also measured *recall* which is “the proportion of known interactions that were present in the list of recommendations over all positive interactions $[py]$ made by a particular user” (Pizzato et al.).

$$Recall(x, R) = \frac{|\{pr : pr \in R, pr \text{ replied positively to } x\}|}{|\{py : py \in M_x, py \text{ replied positively to } x\}|} \quad (6)$$

For example, if user x had a list of the top 5 recommendations where user x messaged only two potential partners within the recommended list, then RECON’s algorithm will only calculate the success rate and recall based on these two interactions. Consider a scenario where user x obtains ten positive replies within the testing period such that $py = 10$ where only two known interactions occur meaning $kr = 2$. If only one of those interactions is positive $pr = 1$, the success rate would be 50% and recall would be 10%.

2.4 Results

The article presents its findings by comparing the success rate and recall for RECON against a non-reciprocal recommender system with varying sizes of the number of recommendations. A baseline success measure of 17.3% was determined by the proportion of messages sent by users that received a positive reply within the testing set [2] without receiving recommendations from any recommendation system.

Given a list of top 10 recommendations, the reciprocal recommender has a 42% success rate and 4% recall whereas the non-reciprocal recommender has a 23% success rate and 1% recall. This trend in improvement of success rate and recall can be seen for varying sizes of recommendation lists thus providing evidence that reciprocity is essential to recommender systems which require two individuals to have a mutual interest.

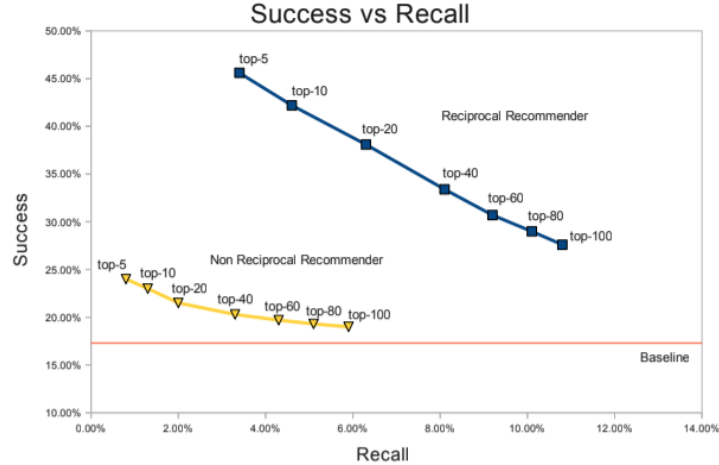


Figure 3: Comparing the success rate and recall between reciprocal recommender, non-reciprocal recommender, and the use of neither [2].

3 Peng Xia et al’s Reciprocal Recommendation System for Online Dating²

In “Reciprocal Recommendation System for Online Dating” (Peng Xia et al), the authors develop a recommendation system based off of RECON that utilizes both content-based and collaborative filtering-based algorithms to measure compatibility between users. The goal of the recommendation system is to recommend users who have a mutual interest as well as a higher likelihood of messaging each other. The design of this recommendation system is based in four components: extraction of user based features, extraction of graph based features, computation of similarity and reciprocal scores, and generation of a recommended user list [3].

3.1 The Design of Recommender System

3.1.1 Extraction of User Based Features

The user must first create a profile that is comprised of various information such as gender, age, height, weight and other categorical attributes. The recommender system implements the same user preference model and compatibility algorithm described in RECON [2].

In addition, the recommender system includes a second compatibility equation to include “numerical attribute information that are compared between the boundaries of continuous ranges” [3]. This second compatibility equation is included due to RECON’s inability to compare two users that do not fall within the same categorical group. An example of this can be seen when comparing a 19 year old user that falls within the 18-20 range and a 21 year old user that falls within the 20-22. The recommender system modifies RECON’s original equation by representing the maximum absolute difference for an attribute a with a value v among all users when comparing user i and j .

²Equations found in this section were obtained from [3]

$$v_a^* = \max_{i \neq j} |v_a^i - v_a^j| \quad (7)$$

Equation 7 is then used to calculate the value of a specific attribute given v_a^* where $Q_a(x, y)$ represents the value of numeric attributes.

$$Q_a(x, y) = \frac{v_a^* - |v_a^x - v_a^y|}{v_a^*} \quad (8)$$

The overall content similarity of user x and user y with their respective attribute sets A_x and A_y is then calculated by following equation:

$$\text{content similarity}(x, y) = \frac{\sum_{a \in A_x \cap A_y} Q_a(x, y)}{|A_x \cap A_y|} \quad (9)$$

3.1.2 Extraction of Graph Based Features

The online dating site used within this study can be represented as a bipartite network between men and women. Graph-based similarity features are then derived from the bipartite graph to indicate a user's active level in dating and attractiveness [3].

The recommender system calculates interest similarity and attractiveness similarity through message history between users. The article defines the set $Se(x)$ as the set of messages user x has sent to user y .

$$Se(x) = \{y : x \text{ has sent a message to } y\} \quad (10)$$

The set $Re(x)$ is defined as the set of messages user x has received from user y .

$$Re(x) = \{y : x \text{ has received a message to } y\} \quad (11)$$

From these two sets, we can define the interest similarity between two same-sex users x and y as the fraction of users both x and y messaged over the set of all users that x and y messaged.

$$\text{interest similarity}(x, y) = \frac{|Se(x) \cap Se(y)|}{|Se(x) \cup Se(y)|} \quad (12)$$

Attractiveness similarity can be defined as the fraction of the set of users that both users x and y received messages from over the set of all users that messaged either user x or y .

$$\text{attractiveness similarity}(x, y) = \frac{|Re(x) \cap Re(y)|}{|Re(x) \cup Re(y)|} \quad (13)$$

3.1.3 Computation of Similarity and Reciprocal Scores

A reciprocal score of two users is calculated using the similarity equations derived from the user-based and graph-based features. The algorithm illustrated in **Figure 4** demonstrates how a reciprocal score is calculated. Within this algorithm, compatibility scores are calculated based on the similarity of neighbor sets that the user is associated with. $Neighbor_1()$ illustrates the set of users that x has sent messages to whereas $Neighbor_2$ illustrates the set of users that y has sent messages to. In addition, $Similarity_1(.)$ and $Similarity_2(.)$ are functions that determine overall similarity between a pair of users. The algorithm calculates $s(x, y)$ as the measure of similarity

between user x and user y where the reciprocal score is the harmonic mean of both $s(x, y)$ and $s(y, x)$.

3.1.4 Generate Recommended User List

For every user within the online dating site, a recommendation list is created that ranks reciprocal scores for top- N users within a list.

Algorithm 1 Reciprocal Score(x, y)

Input: service user x , and recommended user y
Output: reciprocal score

begin
 /* initialize compatible scores s^* */
 $s(x, y) = 0.0$
 $s(y, x) = 0.0$
 /* compute compatible scores for both x and y */
 foreach u in $Neighbor_1(y)$:
 $s(x, y) = s(x, y) + Similarity_1(x, u)$
 end for
 foreach v in $Neighbor_2(x)$:
 $s(y, x) = s(y, x) + Similarity_2(y, v)$
 end for
 /* normalize compatible scores for both x and y */
 if $|Neighbor_1(y)| > 0$
 $s(x, y) = \frac{s(x, y)}{|Neighbor_1(y)|}$
 if $|Neighbor_2(x)| > 0$
 $s(y, x) = \frac{s(y, x)}{|Neighbor_2(x)|}$
 /* compute reciprocal score */
 if $s(x, y) > 0$ and $s(y, x) > 0$
 return $\frac{2}{\frac{1}{s(x, y)} + \frac{1}{s(y, x)}}$
 else
 return 0.0
end

Figure 4: An algorithm that inputs a user x and recommended user y and returns the reciprocal score of a pairing. [3].

4 Analysis

Within the article, the recommender system is evaluated by "comparing the top- N users in the recommended list with the receivers contacted by the service user in test set." [3]. There are three sets of potential partners defined per user. Set T is the set of potential partners recommended to the user, set I is the set of potential partners messaged by the user, and set R is the set of potential partners that have been messaged by the user. From these sets, the authors define I-Precision and I-Recall to "measure an algorithm's performance in recommending users that the service user is interested in and thus likely to contact" [3].

$$I - Precision = \frac{|I \cap T|}{T}, I - Recall = \frac{|I \cap T|}{I} \quad (14)$$

Whereas R-Precision and R-Recall “measure an algorithm’s performance in recommending users who have mutual interest with the service user and are thus likely to reciprocate when contacted” [3].

$$R - Precision = \frac{|R \cap T|}{T}, R - Recall = \frac{|R \cap T|}{R} \quad (15)$$

5 Conclusion

Both articles exemplify viable recommender systems that provide functional top-N list of recommendations for a specific user. Although RECON illustrates a fundamental basis of reciprocity, the RECON system is limited in its implementation due to its inability to compare attributes of different discretised groups. Through the preservation of values of numeric attributes within the recommender system outlined by Peng Xia et al, the content based algorithm posed by Peng Xia et al outperforms RECON’s algorithm in precision and recall. For simplification purposes, we will define RECON’s content based algorithm as CB1 and Peng Xia et al’s content based algorithm as CB2.

Figure 5 and Figure 6 display the performance in comparing CB1 and CB2 in recommending potential partners that the user is interested in.

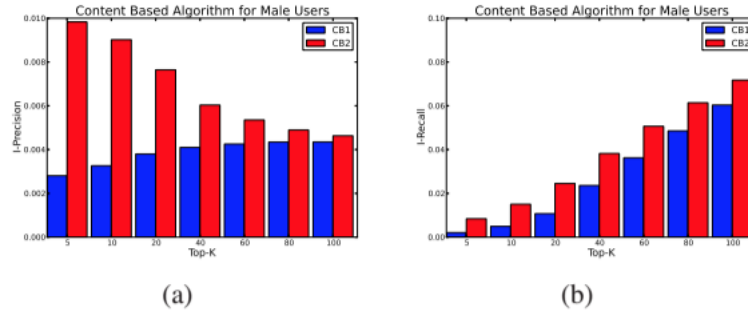


Figure 5: Comparison of I-Precision and I-Recall between CB1 and CB2 for males [3].

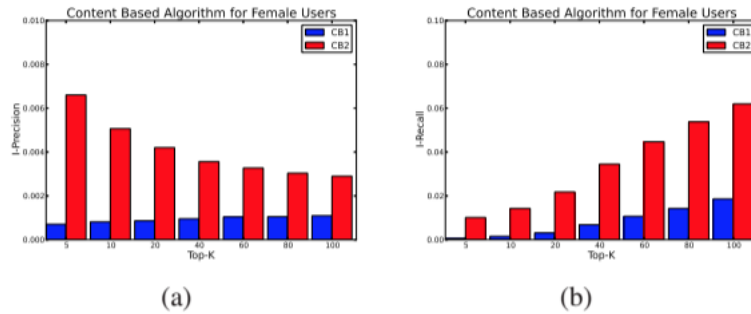


Figure 6: Comparison of I-Precision and I-Recall between CB1 and CB2 for females. [3].

Figures 6 and 7 illustrate the performance of comparing CB1 and CB2 in recommending potential partners who will be messaged by the user.

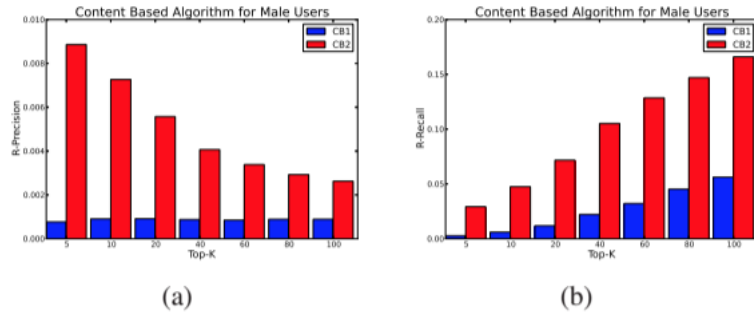


Figure 7: Comparison of R-Precision and R-Recall between CB1 and CB2 for males[3].

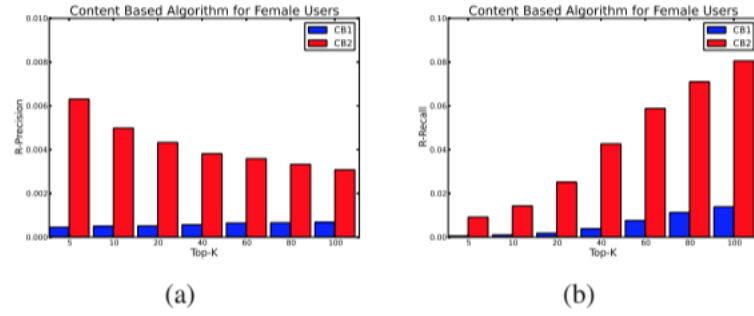


Figure 8: Comparison of R-Precision and R-Recall between CB1 and CB2 for females [3].

References

- [1] Luiz Pizzato, Tomasz Rej, Joshua Akehurst, Irena Koprinska, Kalina Yacef, and Judy Kay. Recommending people to people: the nature of reciprocal recommenders with a case study in online dating. *User Modeling and User-Adapted Interaction*, 23(5):447–488, Nov 2013.
- [2] Luiz Pizzato, Tomek Rej, Thomas Chung, Irena Koprinska, and Judy Kay. Recon: A reciprocal recommender for online dating. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 207–214, New York, NY, USA, 2010. ACM.
- [3] Peng Xia, Benyuan Liu, Yizhou Sun, and Cindy Chen. Reciprocal recommendation system for online dating. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ASONAM '15, pages 234–241, New York, NY, USA, 2015. ACM.