User-to-User Recommendation Using The Concept of Movement Patterns: A Study Using a Dating Social Network

Mohammed Al-Zeyadi, Frans Coenen and Alexei Lisitsa
Department of Computer Science, University of Liverpool, Ashton Building,Ashton Street, Liverpool L69 3BX, UK.
{m.g.a.al-zeyadi, Coenen, lisitsa }@liverpool.ac.uk

Keywords: Movement Pattern Mining, Social Networks, Recommender Systems.

Abstract: Dating Social Networks (DSN) have become a popular platform for people to look for potential romantic partners. However, the main challenge is the size of the dating network in terms of the number of registered users, which makes it impossible for users to conduct extensive searches. DSN systems thus make recommendations, typically based on user profiles, preferences and behaviours. The provision of effective User-to-User recommendation systems have thus become an essential part of successful dating networks. To date the most commonly used recommendation technique is founded on the concept of collaborative filtering. In this paper an alternative approach, founded on the concept of Movement Patterns, is presented. A movement pattern is a three-part pattern that captures the "traffic" (messaging) between vertices (users) in a DSN. The idea is that these capture the behaviour of users within a DSN while at the same time capturing the associated profile and preference data. The idea has been built into a User-to-User recommender system, the RecoMP system. The system has been evaluated, by comparing its operation with a collaborative filtering systems (the RecoCF system), using a data set from the Chinese Jiayuan.com DSN comprising $548,395$ vertices. The reported evaluation demonstrates that very successful results can be produced, a best average F-score value of 0.961.

1 INTRODUCTION

Dating Social Networks (DSNs) have become an impotent platform for people looking for potential partners online. According to a recent survey[1], conducted in the USA, more than 49 million single people (out of 54 million) have used DSNs such as eHarmony and Match.com. Moreover, according to the same survey, 20% of current committed relationships began online. In global terms, Badoo[2] has become the world's largest dating network with more than 346 million registered users and about 350 million messages sent per day. In a large dating network finding potential partners is time consuming, therefore many DSNs give compatible partner suggestions; in the same manner as more general recommender systems, see for example (Resnick and Varian, 1997). Recommender systems have been found to provide significant impact with respect to improving user satisfaction in online retail settings (Sohail et al., 2013; Wang and Wang, 2014). In contrast, developing a recommender system for a DSN is more challenging because the recommender system must satisfy the preferences of pairs of users (Pizzato et al., 2010) as opposed to single users. In this paper, we propose a recommendation system based on the concept of frequently occurring Movement Patterns (MPs).

The MP concept was first proposed in (Al-Zeyadi et al., 2016). An MP is a three part pattern, extracted from a graph, comprising a descriptions of: a "from vertex", a "to vertex" and a connecting edge. The idea was originally proposed in the context of analysing "traffic movement" (real or virtual) in networks, such as freight distribution networks, social networks and computer networks, where the edges represent traffic. The idea being to model "traffic movement" within a network using the idea of frequently occurring MPs and then to use these models to predict future movements. This paper makes the observation that the MP concept can equally well be applied in the context of recommender systems, more specifically recommender systems embedded into DSNs. If we conceive of a DSN as a collection of vertices, each representing an individual, the interchange of messages between vertices can then be considered to represent the traffic (edges) between vertices. Frequently occurring MPs can then be extracted and used to generate recommen-

[1] see http://www.statisticbrain.com/online-dating-statistics/
[2] https://team.badoo.com

dations (to existing users and new users).

Given the above, the main contribution of this paper is an analysis of the usage of the MP concept in the context of recommender systems. More specifically an algorithm, the RecoMP algorithm, is proposed whereby, given a candidate user, a set of "recommendations" can be made using extracted MPs. The utility of the mechanism is illustrated in the context of a DSN where the requirement is that the recommendations most focus on pairs of users (rather than single users as is the case of more standard recommender systems). RecoMP was evaluated using a real-world DSN dataset comprised of $344,552$ males and $203,843$ female users (thus 548,395 vertices in total), and around 3.5 million edges. The evaluation was conducted by comparing the proposed MP based RecoMP algorithm with a benchmark algorithm founded on the concept of collaborative filtering (Schafer et al., 2007). The results demonstrated that the proposed approach produced much better recommendations than the RecoCF comparator algorithm, total average precision, recall and FI values of 0.93, 1.00 and 0.96 were recorded, compared to 0.32, 0.74 and 0.39.

2 Literature Review

In the era of big data the prevalence of social networks of all kinds has grown dramatically, which in turn has led to significant user information overload. Coinciding with this growth is a corresponding desire to analyse (mine) such networks, typically with a view to some social and/or economic gain. Typical tasks are the identification of interacting communities (Oh et al., 2014), and the identification of "influencers" and "followers" (Li et al., 2014). In the context of the work presented in this paper the monitoring of traffic in dynamic networks is of relevance (Al-Zeyadi et al., 2016; Al-Zeyadi et al., 2017). The idea is to predict the future behaviour of a related network (or the same network) according to the current behaviour exemplified in the network being considered. In (Al-Zeyadi et al., 2016) the concept of Movement Patterns (MPs) was proposed, as already introduced in the previous section. In (Al-Zeyadi et al., 2017) the MP concept was used to analyse the databases associated with the UKs Cattle Tracking System managed by the UKs Department for Environment, Food and Rural Affairs (DEFRA). The database records the movement of all cattle between pairs of locations in GB. These locations were viewed as vertices in a network, and the cattle movements as edges between vertex pairs. The database was used to generate a collection of time stamped networks where, for each network, the vertices represented cattle holding areas and the edges occurrences of cattle movement (traffic). The evaluation reported on in (Al-Zeyadi et al., 2017) indicated that MPs could be effectively used to predict traffic movement in previously unseen networks.

Information overload is also of concern in online retail applications where the user is unable to assimilate the wide range of information available concerning products and services. As a consequence the solution adopted by the online retail industry is to make recommendations using what are known as recommendation or recommender systems (Resnick and Varian, 1997). Broadly, recommendation systems can be categorised as being either: ***I*tem-to-Item** or ***U*ser-to-User**. The main different being that User-to-User recommendation systems need to make reciprocal recommendations (Pizzato et al., 2013). Well known examples of Item-to-Item recommendation systems are those embedded in Amazon, Netflix and Spotify; we are all familiar with the "users who bought X also bought Y" mantra. Well known examples of User-to-User recommendation systems are those embedded in Facebook and Linkedin; the "people you might know" mantra. Another example application domain where User-to-User recommender systems are used is Dating Social Networks (DSNs). Dating Networks have become an impotent tool used by people looking for potential romantic partners online; for example, as already noted above, the Badoo DSN has over 340 million registered users.

There has been much work directed at User-to-User recommendation. Of key concern is the quality of the recommended matches; poor quality matching will result in people looking elsewhere. In the context of DSNs Matching is typically done using either: (i) user profiles, (ii) expressed preferences or (iii) user behaviour. For example in (Kunegis et al., 2012) the authors propose a way of modelling both the duality of users similar to each other and preferences towards other users, by using split-complex numbers. The authors demonstrated firstly that their unified representation was capable of modelling both notions of relations between users in a joint expression and secondly that their system could be applied in the context of recommending potential partners. In (Xia et al., 2016) the authors introduced a recommendation system that made use of profiles and references, and provided a list of recommendations that a user might be compatible with by computing a reciprocal score that measured the compatibility between a user and each potential dating candidate. In (Tu et al., 2014), the authors proposed a DSN recommendation framework founded on a Latent Dirichlet Allocation (LDA) model that learns user preferences from observed user messaging behaviour and user profile features. However, the majority of User-to-User DSN recommendation systems are founded on (graph based) Collaborative Filtering (CF) algorithms (Tu et al., 2014; Krzywicki et al., 2014) that focuses on user behaviour. The intuition is that user behaviour is a much better indicator for recommendations than user profiles or expressed preferences (Krzywicki et al., 2014). Examples where CF filtering has been used in the context of DSNs can be found in (Cai et al., 2010; Kutty et al., 2014). Given the popularity, and claimed benefits, of the CF approach this is the approach with which the proposed MP based RecoMP algorithm is compared. For the purpose of the evaluation the authors developed a bespoke CF based DSN recommendation algorithm called RecoCF, this is described in further detail in Section 5.

The distinguishing feature between the above DSN recommender systems and the DM based system proposed in this paper is the MP concept. To the best of the authors' knowledge there has been no work directed at user-to-user recommendation using MPs as presented in this paper. There has of course been plenty of work directed at finding patterns in data. The earliest examples are the Frequent Pattern Mining (FPM) algorithms proposed in the early 1990s (Agrawal et al., 1994). The main objective being to discover sets of attribute-value pairings that occur frequently which can then be used to formulate what are known as association rules which in turn have been used for recommendation purposes, examples can be found in (Sandvig et al., 2007; Lin et al., 2002). A frequently quoted disadvantage of FPM is the significant computation time required to generate large numbers of patterns (many of which may not even be relevant). The MP Mining (MPM) concept presented in this paper shares some similarities with the concept of FPM. However, the distinction between movement patterns and traditional frequent patterns is that movement patterns are more prescriptive, as will become apparent from the following section. Note also that the movement patterns of interest with respect to this paper are traffic movement patterns and not the patterns associated with the video surveillance of individuals, animals or road traffic; a domain where the term "movement pattern" is also sometimes used.

3 System Overview

An overview of the proposed MP based DSN recommendation systems is presented in this section. The section commences, Sub-section 3.1, with a review of the basic operation of DSN systems. A formalism for the MP concept is then presented in Sub-section 3.2, followed by a formal definition of the problem domain and a problem statement in Section 3.3.

3.1 DSN Application Framework

The basic operation of DSNs (see Figure 1), regardless of the adopted recommendation system used, is as follows.

1. ***J*oining the network**. When a new user joins a DSN a new user profile is created using information provide by the new user; information such as: age, gender, location, job, education, income, smoking, drinking, religion, hobbies, and so on.
2. ***B*rowsing**. After the creation of the profile the new user can browse the profiles of existing users (as can existing users).
3. ***O*ne sides match**. While browsing, users may send messages to other users.
4. ***R*eciprocal match**. On receipt of a massage a user can return a message (reciprocate). Where this happens an edge is established in the DSN. The strength of an edge can be defined in terms of the quantity and/or duration of the messages. A degradation factor can also be applied to take into account the temporal nature of the network.

Given the large number of users, browsing is unlikely to be successful, hence DSN systems also provide recommendations. Recommendations can be made when a new user joins the network and periodically for existing users. As already noted, the most commonly adopted techniques for making recommendations are founded on some form of Collaborative Filtering.

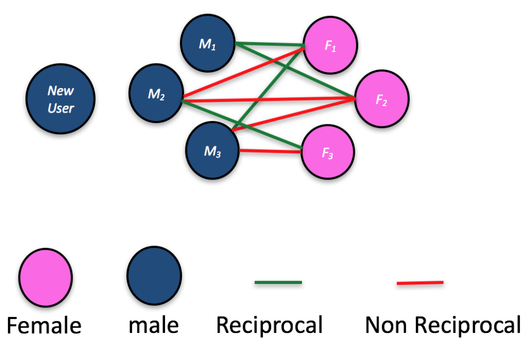

Figure 1: Example Dating Network.

3.2 Movement pattern formalism

From the foregoing we are interested in building a recommender system for a DSN system founded on the concept of MPs. In the introduction to this paper it was noted that a MP is a three-part pattern. More formally a MP comprises a tuple of the form:

$$\langle F, E, T \rangle \tag{1}$$

where F, E and T are sets of attribute values. More specifically the attribute value set F represents a "From" (sender) vertex, T a "To" (receiver) vertex, and E an "Edge" connecting the two vertices describing the nature of the traffic (details of movement) between them. We refer to a tuple of this type using the acronym FET. The minimum number of attribute values in each part (set) must be at least one. The maximum number of values depends on the size of the attribute sets to which F, E and T subscribe, an MP can only feature a maximum of one value per subscribed attribute. The attribute set to which F and T subscribe is given by $A_V = \{\phi_1, \phi_2, \dots\}$, whilst the attribute set for E is given by $A_E = \{\varepsilon_1, \varepsilon_2, \dots\}$. Note that F and T subscribe to the same attribute set because they are both movement network vertices, and every vertex (at least potentially) can be a "from" or a "to" vertex in the context of MPM. Each attribute in A_V and A_E also has a value domain associated with it.

Any given network can also be represented as a set of tuples of the form $\langle F, E, T \rangle$ (Equation 1). In other words a given network can be encapsulated in the form of a dataset $D = \{F_1, F_2, \dots\}$, where each $F_i \in D$ is a FET. An MP is then a FET that occurs frequently in D, where frequency is defined in terms of a frequency threshold σ, a percentage value between 0.0 and 100.0 indicating the proportion of the number of occurrences of a particular MP with respect to the total number of records (edges) in the data set, or data set segment, under consideration. In the context of DSNs the sets F and T represent DSN user profiles, while the set E represents the nature of the reciprocal messaging between users. A MP is then a frequently occurring FET that encompasses a pair of user profiles and reciprocal messaging behaviour. Further details concerning MPs and FETs can be found in found in (Al-Zeyadi et al., 2016) and (Al-Zeyadi et al., 2017).

3.3 Problem Statement

In the context of the work presented in this paper a dating network G is defined in terms of a tuple of the form $\langle V, E \rangle$, where V is a set of vertices representing the users of the DSN and E is the set of edges representing reciprocal communication between users. Each vertex $v_i \in V$ is defined by a set of attribute values representing the profile of the user. In the case of the dataset used for the evaluation purposes, as reported on later in this paper, 25 different attribute values were used to describe users profiles. Each edge $e_i \in E$ is then defined by a another set of attribute values describing the nature of the communication. For the evaluation considered later in this paper only two edge attribute was considered, "communication type" and "number of messages sent", the first had two potential values: Reciprocal and Non reciprocal. The second had a range of values.

4 Recommendation System Based on Movement Patterns (RecoMP)

In this section the proposed MP based DSN recommendation algorithm is presented, the RecoMP algorithm. Recall that the idea is to use knowledge of existing frequently occurring MPs in the DSN to make recommendations. A particular challenge of finding frequently occurring MPs in DSNs is the size of the networks to be considered. The exemplar dataset used for the evaluation reported on later in this paper comprised 548,395 vertices and some 3.5 million edges. In other words we cannot mine and maintain all the MPs that might feature in the data set. Note that although the number of MPs generated can be reduced by using a high σ threshold this is undesirable as we need to use a low σ threshold so as to ensure no significant MPs are missed (the most appropriate value for σ will be considered in Section 6). The solution is to mine MPs as required with respect to a specific user and to consequently generate recommendations with respect to that specific user. Users would be considered in turn, but recommendations would be made periodically. It would therefore not be necessary to consider all DSN users in one processing run. In addition, by mining MPs on a required basis, the continuously evolving (dynamic) nature of DSNs can be taken into account.

The pseudo code for the RecoMP process is presented in Algorithm 1. The inputs are: (i) a given user profile u_{new}, (ii) the set of all user profiles U, (iii) the DSN represented as a dataset D comprised of a set of FETs (as described above), and (iv) a desired support threshold σ. Note that for illustrative purposes, in Algorithm 1, we have assumed a new user, but this could equally well be an existing user for whom a new set of recommendations is to be generated. The output is a set R of recommended users (matches). Inspection of the algorithm indicates that it comprises two sub-processes: (i) ***M*ining** (lines 7 to 21) and (i) ***R*ecommendation** (lines 22 to 28).

The mining sub-process is where the relevant MPs are generated. MPs are stored in a set $M = \{\langle MP_1, count_1 \rangle, \langle MP_1, count_1 \rangle, \dots\}$. On start up (line

Input:
1 u_{new} = new joined user profile vector
2 U = Collection of all user profile vectors
3 D = Collection of FETs $\{r1, r2, ...\}$ describing network G
4 σ = Support threshold
Output:
5 R = Set of recommended users
6 Start:
7 ***M*ining Part:**
8 M = $\emptyset$
9 D_{new} = Pruning D by looping through D and considering only FET_i where F or T similar to u_{new}
10 $ShapeSet$ = the set of possible shapes $\{shape_1, shape_2, \ldots\}$
11 **forall the** $shape_i \in ShapeSet$ **do**
12   **forall the** $r_j \in D_{new}$ **do**
13   **if** r_j *matches* $shape_i$ **then**
14   MP_k = MP extracted from r_j
15   **if** MP_k *in* M **then** increment support
16   **else** $M = M \bigcup \langle MP_k, 1\rangle$
17   **end**
18 **end**
19 **forall the** $MP_i \in M$ **do**
20   **if** *count for* $MP_i < \sigma$ **then** remove MP_i from M
21 **end**
22 ***R*ecommendation Part:**
23 **forall the** $u_i \in U$ **do**
24   **forall the** $MP_j \in M$ **do**
25   **if** $u_i \subseteq MP_j$ *and* $u_i \nsubseteq R$ **then**
26         $R = R \bigcup u_i$
27   **end**
28 **end**

Algorithm 1: The RecoMPA Algorithm

8) M is set to the empty set $\emptyset$. The sub-process commences (line 8) by pruning D to create D_{new} ($D_{new} \subset D$) so that we are left with a set of FETs where either the From and/or the To part correspond (are similar) to u_{new}. This benefit of this pruning is that it results in a significantly reduced search space. Similarity measurement was conducted using the well known Cosine similarity metric calculated as shown in Equation 2 where A and B are the set of attribute values of a newly joined user, and a selected user in the network, respectively.

$$Similarity = cos(\Theta) = \frac{\sum_{i=1}^{n} A.B}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (2)$$

Next (line 9) a "shape set" is generated to support MP extraction. A shape is a MP template (prototype) with a particular configuration of attributes taking from the attribute sets A_V and A_E without considering the associated attribute values. Once generated shapes can be populated with attribute values to give candidate MPs. The idea is to enhance the efficiency of calculating MPs by considering potential MPs in terms of the attributes they might contain, as oppose to the individual attribute values they might contain, given that size of the set of attributes will be less than the size of the concatenated set of attribute values. The maximum number of shapes that can exist in D_{new} is given by Equation 3, where $|A_v|$ and $|A_E|$ are the number of vertex and edge attributes that feature in D_{new}.

$$(2^{|A_v|} - 1) \times (2^{|A_E|} - 1) \times (2^{|A_v|} - 1) \quad (3)$$

Returning to algorithm 1 the next step is to populate the set of generated shapes (lines 11 to 18). For each shape $shape_i$ in the shape set, and for each FET (record) r_j in D_{new}, if r_j matches $shape_i$ r_j it is temporarily stored in a variable MP_k. Note that a record r_j matches a $shape_i$ if the attributes featured in the shape also feature in r_j. If MP_k is already contained in M we increment the associated count (line 15), other wise we add MP_k to M with a count of 1. Once all shapes have been processed we loop through M (lines 19 to 20) and remove all MPs whose support count is less than σ.

When the set of frequently occurring MPs has been generated the recommender sub-process is commenced (line 22). For each MP MP_j in M, and each user profile (vertex) u_i in U, if u_i is a subset of either the From or To part of MP_j, and has not previously been recorded in R, u_i is appended to R (line 26). In this manner a set of recommended users is generated.

Note that shape based approach to MP mining described above lends itself to parallelisation. Each shape can be populated and the various resulting MP instances counted on a separate processing unit without requiring any messaging between units. Technologies such as Map Reduce (MR) on a top of Hadoop (Dean and Ghemawat, 2008) or the well known Massage Passing Interface (MPI) (Gropp et al., 1999) would be appropriate here as discussed in (Al-Zeyadi et al., 2017).

5 Recommendation System Based on Collaborative Filtering (RecoCF)

To evaluate the proposed RecoMPA algorithm described above a benchmark algorithm was required. As noted in Section 3, the majority of User-to-User DSN recommendation systems are founded on (graph based) Collaborative Filtering (CF) approaches (Tu et al., 2014; Krzywicki et al., 2014). A benchmark CF based DSN recommendation algorithm was therefore developed, the RecoCF algorithm. The general methodology of Collaborative Filtering, for any system, can be described in two steps:

1. Identify users who share the same vector pattern with the service user (the user whom the prediction is for).
2. Use the preferences of those users founded in step 1 to create a prediction (recommendation) for the service user.

The same methodology was adopted with respect to the purpose built CF based DSN recommendation RecoCF algorithm. The pseudo code for the RecoCF algorithm is presented in Algorithm 2. As in the case of RecoMP algorithm, the RecoCF algorithm takes the same input except there is no need for a σ threshold. The output, as before, is a set of recommended users R. The algorithm commences (line 6), as in the case of the RecoMP algorithm, by pruning the dataset D to give D_{new}. Then for all records (FETs) in D_{new} the From and To attribute value sets are extracted (lines 8 and 9), the sets $From_i$ and To_i. If $From_i$ is a subset of u_{new} (the new user profile) the user profile associated with $From_i$ is added to R if it has no already been included. Similarly if To_i is a subset of u_{new} the user profile associated with To_i is added to R, again provided if has not already been included. The result is a set R of recommended users (matches).

Input:
1 u_{new} = new joined user profile vector
2 U = Collection of all user profile vectors
3 D = Collection of FETs $\{r1, r2, ...\}$ describing network G
Output:
4 R = Set of recommended users
5 Start:
6 D_{new} = Pruning D by looping through D and considering only FET_i where F or T similar to u_{new}
7 **forall the** $D_i \in D_{new}$ **do**
8 $From_i$ = return From part from D_i
9 To_i = return To part from D_i
10 **if** $From_i \subseteq u_{new}$ *and* $From_i \nsubseteq R$ **then**
11 $R = R \bigcup To_i$
12 **else if** $To_i \subseteq u_{new}$ *and* $To_i \nsubseteq R$ **then**
13 $R = R \bigcup From_i$
14 **end**

Algorithm 2: The RecoCF Algorithm

6 Evaluation

This section reports on the evaluation conducted with respect to the proposed RecoMP algorithm. The evaluation was conducted using a FET database extracted from a dataset obtained from the Jiayuan.com DSN. The objectives of the evaluation were to compare the operation of the proposed MP based RecoMP algorithm in comparison with standard Collaborative Filtering (the RecoCF algorithm from Section 5). The metrics used for the evaluation were: (i) Recall (R), (ii) Precision and (iii) F-score (F).

Table 1: TCV results using the RecoMP algorithm

Tenth	RecoMP		
	P	R	F
# 1	0.938	1.000	0.978
# 2	0.908	1.000	0.949
# 3	0.917	1.000	0.956
# 4	0.948	1.000	0.972
# 5	0.948	1.000	0.972
# 6	0.948	1.000	0.972
# 7	0.928	1.000	0.961
# 8	0.928	1.000	0.961
# 9	0.952	1.000	0.974
# 10	0.867	1.000	0.917
Avarage	**0.928**	**1.000**	**0.961**
SD	**0.02**	**0.00**	**0.02**

Table 2: TCV results using the RecoCF algorithm

Tenth	RecoCF		
	P	R	F
# 1	0.217	0.764	0.298
# 2	0.369	0.831	0.470
# 3	0.325	0.760	0.416
# 4	0.305	0.722	0.364
# 5	0.305	0.722	0.364
# 6	0.305	0.722	0.364
# 7	0.333	0.756	0.424
# 8	0.354	0.763	0.416
# 9	0.265	0.683	0.361
# 10	0.446	0.717	0.439
Avarage	**0.322**	**0.744**	**0.392**
SD	**0.058**	**0.038**	**0.048**

6.1 Data Sets

For the conducted evaluation reported on in this paper a dataset was obtained from Jiayuan.com[3]. Jiayuan.com is the most popular DSN in China; in 2011 it was reported t have 40.2 million subscribers (users), and 4.7 million active monthly subscribers. The data obtained comprised $548,395$ users ($344,552$ men and $203,843$ women) and details concerning whether a user had messaged another (no information quantifying the messaging activity was available). Each user

[3] http://www.jiayuan.com

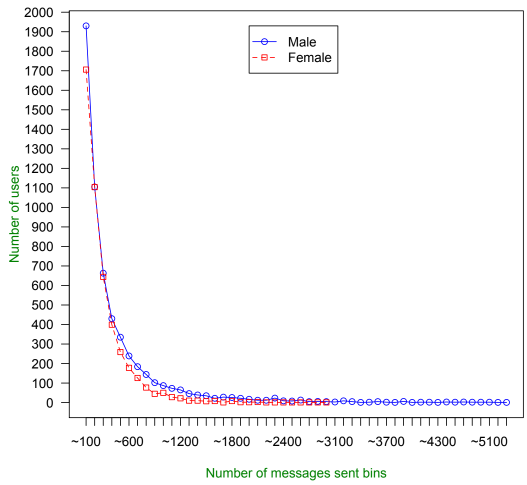

Figure 2: Male and Female Normal Distribution.

had a profile and a set of preferences associated with it. Unlike European or US DSNs, Jiayuan.com, in line with other Chinese DSNs, is directed at the (heterosexual) marriage market rather than the shorter term relationship market, and thus user profiles tend to reflect this; profiles comprise: age, height, education, location, occupation, place of work, income, home ownership, car ownership and so on. Preferences include things like: age range, height range, education and location. The data set was processed firstly so that each user was defined by a set of 25 (profile and preference) attributes, thus $|A_v| = 25$. It was then processed again so as to generate a network where the vertices represented users. Edges where included wherever two users had messaged each other, in other words the messaging was reciprocal, thus $|A_e| = 1$ with only a single value. Unfortunately the nature of the data set was such that we could not extract a more comprehensive edge attribute set. Converting this network into a FET database resulted in a database comprising $3,311,076$ records. The normal distribution of the users' activity, in terms of the number of messages sent, is presented in Figure 2. From the figure it can be seen that the majority of users sent 100 messages over the considered time frame. Given a new user (or an existing user for whom we wish to make a recommendation), if we find a frequent MP within the existing network where either the From or To part matches the description (profile) of the new user we recommend the associated existing users to the new user.

6.2 Performance Effectiveness of RecoMP with respect to RecoCF

To determine the effectiveness of the proposed RecoMP algorithm, in comparison with RecoCF, two sets of experiments were conducted. The comparison was conducted using a variation of Ten Cross Validation (TCV) whereby the entire Jiayuan.com FET database was divided into tenths and the process run ten times with a different tenth used for testing. More specifically for each run a random sample of ten users was extracted from the testing tenth and used for the evaluation. In this manner the process of TCV could be conducted without processing all $548,395$ vertices represented in the database. For both sets of experiments a threshold value of $\sigma = 1.0$ was used.

The results are given in Tables 1 and 2, Table 1 gives the results using the RecoMP algorithm while Table 2 gives the results using the RecoCF algorithm. The tables give the average Precision (P), Recall (R) and F-score (F) for each tenth, and a total average and Standard Deviation (SD).

From the above it can clearly be seen that the recommendations made using the RecoMP algorithm are better than those generated using Collaborative Filtering (the RecoCF algorithm). The total average recall, precision and F-score using RecoMP were 0.928, 1.000 and 0.961; compared to total average recall, precision and F-score values of 0.322, 0.744 and 0.392 using RecoCF with small SD values were recorded. It is also interesting to note that the total average precision using RecoMP, as before, was frequently 1.000; meaning we often make all the correct recommendations and no incorrect recommendations.

7 Conclusion

In this paper, the authors have proposed a recommendation system, directed at Dating Social Networks (DSN), founded on the concept of Movement Patterns (MP), patterns that capture the nature of traffic movement between vertices in networks. The idea is to extract frequently occurring MPs from a current network and use these to inform a User-to-User recommender DSN system. The idea was built into an algorithm, the RecoMP algorithm, and tested by comparing the operation of this algorithm with a Collaborative Filtering approach, RecoCF algorithm. For the evaluation a large network, extracted from Jiayuan.com DSN system, comprising 3,311,076 vertices (users) was used. Excellent results were produced, a best total average F-score value of 0.961 was obtained using the RecoMP algorithm compared to a value of 0.392 using the RecCF algorithm. However, for general applicability to large DSN, the efficiency of the approach needs to be improved. A potential avenue for future work is thus to investigate the potential for using some form of parallel processing, for example using the well known Massage Pass Interface (MPI) or Hadoop/MapReduce. One of the advantages offered by the "Shape" based approach to mining MPs, as proposed in this paper, is that it lends itself to parallelisation, potentially each possible shape can be processed using a separate processing unit.

ACKNOWLEDGEMENTS

The authors would like to thank the China University of Science and Technology, and the School of Statistics at the Renmin University of China Statistical Centre, for providing the jiayuan.com dataset used for evaluation purposes in this paper. Also, the first author would like to thank the Iraqi Ministry of Higher Education and Scientific Research, and University of Al-Qadisiyah, for funding this research.

REFERENCES

Agrawal, R., Srikant, R., et al. (1994). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499.

Al-Zeyadi, M., Coenen, F., and Lisitsa, A. (2016). Mining frequent movement patterns in large networks: A parallel approach using shapes. In *Research and Development in Intelligent Systems XXXIII: Incorporating Applications and Innovations in Intelligent Systems XXIV 33*, pages 53–67. Springer.

Al-Zeyadi, M., Coenen, F., and Lisitsa, A. (2017). On the mining and usage of movement patterns in large traffic networks. In *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on*, pages 135–142. IEEE.

Cai, X., Bain, M., Krzywicki, A., Wobcke, W., Kim, Y. S., Compton, P., and Mahidadia, A. (2010). Learning collaborative filtering and its application to people to people recommendation in social networks. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 743–748. IEEE.

Dean, J. and Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

Gropp, W., Lusk, E., and Skjellum, A. (1999). *Using MPI: portable parallel programming with the message-passing interface*, volume 1. MIT press.

Krzywicki, A., Wobcke, W., Kim, Y. S., Cai, X., Bain, M., Compton, P., and Mahidadia, A. (2014). Evaluation and deployment of a people-to-people recommender in online dating. In *AAAI*, pages 2914–2921.

Kunegis, J., Gröner, G., and Gottron, T. (2012). Online dating recommender systems: The split-complex number approach. In *Proceedings of the 4th ACM RecSys workshop on Recommender systems and the social web*, pages 37–44. ACM.

Kutty, S., Nayak, R., and Chen, L. (2014). A people-to-people matching system using graph mining techniques. *World Wide Web*, 17(3):311–349.

Li, J., Peng, W., Li, T., Sun, T., Li, Q., and Xu, J. (2014). Social network user influence sense-making and dynamics prediction. *Expert Systems with Applications*, 41(11):5115–5124.

Lin, W., Alvarez, S. A., and Ruiz, C. (2002). Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, 6(1):83–105.

Oh, H. J., Ozkaya, E., and LaRose, R. (2014). How does online social networking enhance life satisfaction? the relationships among online supportive interaction, affect, perceived social support, sense of community, and life satisfaction. *Computers in Human Behavior*, 30:69–78.

Pizzato, L., Rej, T., Akehurst, J., Koprinska, I., Yacef, K., and Kay, J. (2013). 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.

Pizzato, L., Rej, T., Chung, T., Koprinska, I., and Kay, J. (2010). Recon: a reciprocal recommender for online dating. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 207–214. ACM.

Resnick, P. and Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3):56–58.

Sandvig, J. J., Mobasher, B., and Burke, R. (2007). Robustness of collaborative recommendation based on association rule mining. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, pages 105–112, New York, NY, USA. ACM.

Schafer, J., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. *The adaptive web*, pages 291–324.

Sohail, S. S., Siddiqui, J., and Ali, R. (2013). Book recommendation system using opinion mining technique. In *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, pages 1609–1614. IEEE.

Tu, K., Ribeiro, B., Jensen, D., Towsley, D., Liu, B., Jiang, H., and Wang, X. (2014). Online dating recommendations: matching markets and learning preferences. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 787–792. ACM.

Wang, X. and Wang, Y. (2014). Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 627–636. ACM.

Xia, P., Zhai, S., Liu, B., Sun, Y., and Chen, C. (2016). Design of reciprocal recommendation systems for online dating. *Social Network Analysis and Mining*, 6(1):1–16.