Social Recommendations for Personalized Fitness Assistance

Saumil Dharia[1] · **Magdalini Eirinaki**[1] · **Vijesh Jain**[1] · **Jvalant Patel**[1] ·
Iraklis Varlamis[2] · **Jainikkumar Vora**[1] · **Rizen Yamauchi**[1]

Abstract Wearable technology allows users to monitor their activity and pursue a healthy lifestyle through the use of embedded sensors. Such wearables usually connect to a mobile application that allows them to set their profile, and keep track of their goals. However, due to the relatively high maintenance of such applications, where a significant amount of user feedback is expected, users who are very busy, or not as self-motivated, stop using them after a while. It has been shown that accountability improves commitment to an exercise routine. In this work we present the PRO-Fit framework, a personalized fitness assistant aiming at engaging users in fitness activities, incorporating a social element. The PRO-Fit architecture collects information from activity tracking devices and automatically classifies their activity type. Moreover, the framework incorporates a social recommender system. Using collaborative filtering on user profile and activity data PRO-Fit generates personalized fitness schedules based on their availability and wellbeing goals. We also incorporate the social network community of the application's users and identify different tie strengths based on the user's connections and location. The output of the recommendation process is twofold, as both new activities, as well as fitness buddies, are being recommended to each user.

Keywords wearable technology · activity tracking · social recommendation · collaborative filtering · personalized assistant

[1]Computer Engineering Department, San Jose State University, San Jose, CA, USA
Tel.: +1 (408) 924-3828
E-mail: magdalini.eirinaki@sjsu.edu ·
[2]Department of Informatics and Telematics, Harokopio University of Athens, Athens, Greece
Tel.: +30 210-9549-405 E-mail: varlamis@hua.gr

1 Introduction

Fitness-related smartphone applications have gained popularity in recent years as they help users integrate health and fitness activities into their daily lives creating better personal health engagement and raise health adherence. The advancements in wearable technology, where embedded accelerometers, gyroscopes, GPS tracking and other sensors enable the users to actively monitor their activity have revolutionized the field, by allowing users to engage in simpler activities, such as walking or running. However, most of the existing technologies are based on an interactive model, expecting from the users to actively keep track of their workouts and other fitness and health goals, leading less self-motivated individuals to soon loose interest and stop exercising. According to a study that reviewed 200 existing health and fitness applications, the main priority for users is to have an application that makes any physical activity entertaining and rewarding, and motivates them to continue striving for achieving targets they set [10]. It is therefore essential for activity tracking and well-being applications to incorporate these two factors to allow people to have a positive attitude towards fitness activities. It is also important to convince them that they have the ability to attain the goals that they have set for themselves. This demand can be met by quantifying and setting accurate goals for each individual.

Sometimes, it is observed that due to lack of company in performing a fitness activity, people get complacent and lose interest. There are many studies that have shown that exercising as part of a group leads to higher exercise adherence [8, 13]. One well-studied phenomenon is the Köhler effect [14, 26] where researchers have shown that people working out on dyads (i.e. pairs) keep each other motivated and exercise longer. This im-

portant element can be partially addressed by embedding rewards systems or credit points in the applications that support the wearable devices, however incorporating social recommendations by suggesting friends with similar interests and goals, would be much more effective.

In this paper, we present PRO-Fit (standing for Personalized Recommender and Organizer Fitness assistant), a personalized fitness assistant framework, that integrates multiple sources of information, including the user's preferences, their calendar, and their social network, and proactively pushes notifications to the user, in an effort to keep her motivated and focused on her goals. Our motivation draws from the fact that for busy individuals, the existing interactive models might not be enough to keep them motivated to engage in fitness-related activities.

The proposed framework incorporates two critical modules: an activity classifier, and a ranking and recommendation engine. We use machine learning algorithms on activity data to build predictive models that classify the user's activity into specific types. We build user profiles reflecting their current lifestyle (e.g. sedentary vs. active), age, weight, goals (e.g. time spent exercising each week), and preferences (e.g. favorite fitness activities, level of intensity etc.). This user profile is fed to a hybrid recommendation system that matches the user's profile to available activities, ranked in terms of similarity, but also taking into consideration the geolocation and time availability.

In addition, PRO-Fit uses collaborative filtering to generate activity and fitness buddy recommendations to users of fitness tracking applications. In a nutshell, the recommendation engine takes as input the user's activity profile, availability and location, and finds similar users. The ones with whom the user has stronger ties (in terms of location, social relationship, and preferences) are ranked higher. The system uses these similarities to identify potential fitness buddies but also to generate new activity recommendations for the user. For instance, PRO-Fit might recommend a 1-hour yoga class at the University fitness center during lunch time for user A, who is employee at the University, and 20-min jogging at the nearest park for user B, who is a student and has 30 minutes between classes. In addition, it might recommend user B as a cardio fitness buddy to user A when both of them are available for one hour during lunch time on a day that both of them are located on campus.

The rest of the paper is organized as follows: In Section 2 we review the related work. In Section 3 we present an overview of the PRO-Fit personalized fitness assistant framework and outline the basic components of the activity classifier and calendar management, introduced as part of our previous work [11,12]. In Section 4 we present in detail the proposed social recommendation system. The experimental evaluation of the social recommendation module is discussed in Section 5. In Section 6 we present the results of a user study conducted with real users of the PRO-Fit prototype application. Finally, we conclude with our plans for future work in Section 7.

2 Related Work

The activity trackers allow for recording various types of data using their inbuilt accelerometers, gyroscopes, GPS and other sensors. The data collected from these devices can be used to determine the type of activity and fitness level of an individual. A lot of research work has been done in this area, by employing machine learning algorithms on past user activity data [20], heart rate data [6], and accelerometer data [4, 5] to identify the type of activity, and/or estimate caloric consumption. In [15], the authors use the phone accelerometer data and the WEKA tool to aggregate raw time series data and generate a predictive model for activity recognition. Contrary to prior work, this paper focuses on using a single device conveniently kept anywhere, rather than multiple devices distributed over the body for tracking purposes. In [16], the authors use an on-body chest sensor in coordination with a smartphone to collect the data for the activities performed by the individual, whether static or dynamic. In [17] the authors propose a fitness architecture named Digital Fitness Connector (DFC), which allows the user to monitor physical activity in real-time as well as post-workout. It captures and stores data collected from health and fitness related sensors. DFC works with smartphone platform but the user has the flexibility in carrying the smartphone or just the DFC. The majority of the proposed approaches employ decision trees and their scalable variations (such as random forests) to perform the activity classification [31], however clustering approaches have also been used to split activity data into categories [24]. While most of existing works focus on improving the activity prediction process, most require extensive user profiling and interaction throughout the day. In our previous work [11] we proposed the use of Gradient Boosted Trees to classify a user's activity, as recorded by a wearable device's accelerometer. We integrated it in an application that scheduled workout sessions based on a user's goals, their preferred activity and their availability, requiring minimal interaction from the user.

While research has shown that exercising in groups or having accountability partners or fitness buddies leads

to higher exercise adherence [13, 14], very few research works exist that have focused on this subject from a technical standpoint. In [9], the authors propose the use of gamification through an application and show that cooperation among users increased the effort put and enhanced the physical activities of participants. In this study, the user connections were virtual (using the app). Other related studies focus mostly on the user experience, through a human-computer interaction and human factors' point of view [2, 19].

In this work, we view this problem from a social recommendation systems' perspective. Tang et al. [30] give a narrow definition of social recommendation as "any recommendation with online social relations as an additional input, i.e., augmenting an existing recommendation engine with additional social signals", while a broader definition, refers to recommender systems targeting social media domains such as blogs and microblogs, multimedia, question answering, job finding and news sites [18]. In all the above domains, typical collaborative filtering (other user preferences) is combined with social based filtering (friends' preferences) and individual filtering (user preferences) in order to improve the quality of recommendations.

In the survey work of Bernardes et al [7], authors conclude that the field of social Recommenders Systems (RS) built on implicit social networks seems particularly promising, propose a social filtering formalism, and with their experiments on music and movie preference datasets, they find that one has to test and try a full repertoire of candidate RS, fine-tune parameters and select the best RS for the performance indicator he/she cares for. Authors in [32] study the efficiency of social recommender networks merging the social graph with the co-rating graph and consider several variations by altering the graph topology and edge weights. With experiments on the Yelp dataset, they conclude that social network can improve the recommendations produced by collaborative filtering algorithms when a user makes more than one connection.

In our work, we consider our recommendation system to be a social one as a) it applies to the social network of the users of the application, but also b) it can integrate social graph-based information to enhance the recommendation process. The literature survey performed so far shows that most works employ existing datasets from music or movie rating networks to experimentally evaluate the models or algorithms proposed, but none of them actually applies the proposed solution to a real-world application. Moreover, despite the positive results attained from the use of social networks in motivating the users of fitness applications to exercise [3, 29], no fitness applications cur-

rently exist that recommend activities to users using a social recommender system. The only recommender system found so far in the healthcare domain that extends the collaborative filtering approach with topic- and sentiment-related information from user-provided reviews is a doctor recommender system called iDoctor [33]. In that work, authors employ the Yelp dataset for extracting reviews and ratings for doctors and evaluating their system. However the incorporation of social information is included in their plans for future work.

In a nutshell, we propose a framework that minimizes the need for user input and allows the end user to keep a healthy, active lifestyle by proactively reminding him/her about their goals and generating personalized fitness recommendations. Contrary to previous work, this framework integrates collaborative filtering to generate activity and fitness buddy recommendations, using as input both its users' activities and schedules, as well as their social network and geo-location. To the best of our knowledge, this is the first work to integrate social recommendations in a personalized fitness assistant framework.

3 PRO-Fit Architecture

The high-level system architecture is shown in Figure 1. The framework's main modules are the *PRO-Fit UI*, the *Activity Tracking & Classification* module, the *Calendar Integration Manager*, and the *Social Recommendation System*. The application collects and generates various types of data; it collects and analyzes accelerometer data, integrates the user's external calendar data, as well as their contacts (address book and social network-derived data), GPS signal, but also stores profile and preference data as input by the user through the application. All the data are stored in the application's data store.

In this Section we provide an overview of the *Pro-Fit UI*, *Activity Tracking & Classification* and *Calendar Integration Manager* modules. For more details the reader may refer to [6, 7]. The *Social Recommendation System* is discussed in detail in Section 4.

3.1 User Interface and PRO-Fit UI

The activity tracking front-end can be any wearable device, including the user's cell phone, as long as it has an accelerometer. The accelerometer sends activity data to the front-end mobile application. The users interact with the mobile application through a dashboard where they can create and update their profile, including rating several types of activities and set their health and
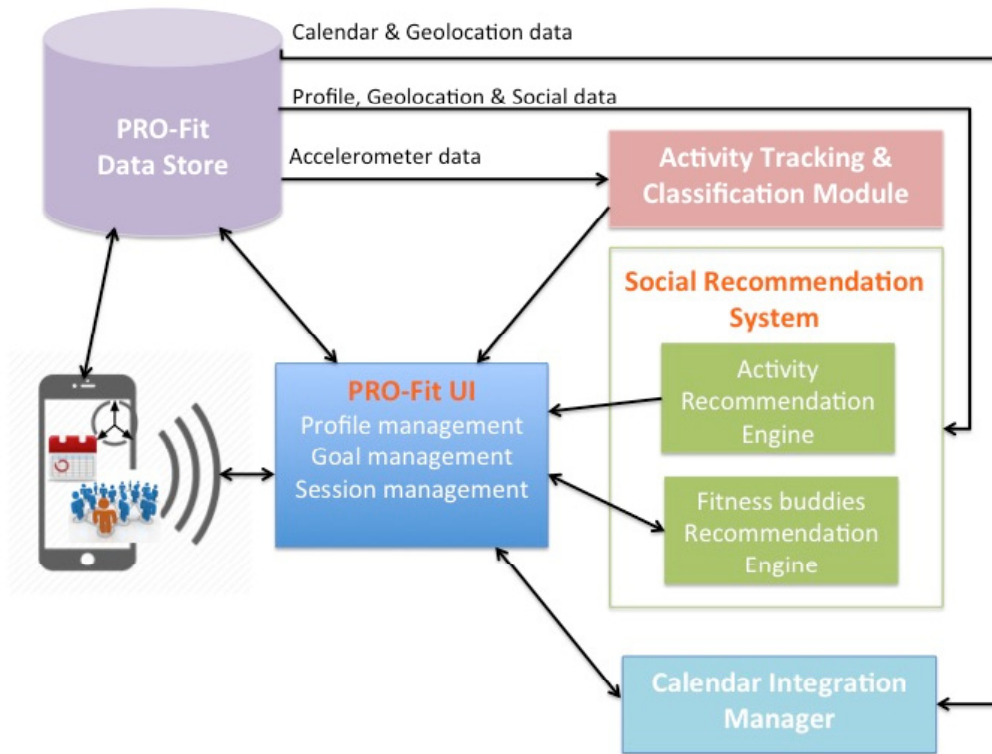
Fig. 1: PRO-Fit architecture

fitness goals. More specifically, once the user logs in to the system for the first time, the system requires from them to build their profile, by providing demographic information (such as age, gender, location, etc.), as well as their health and fitness goals (such as times of exercise per time period t, calories burnt per time period t, etc.). Most importantly, the users are provided with a list of activities (e.g. running, walking, biking, etc.) and are asked to rate them, in a scale of 1 to 10, in terms of how much they enjoy to participate in each activity. This process, handled by the *Profile Manager* and the *Goal Manager*, needs to be performed only once, unless the user wants to update their profile or set new goals. The third component of this module, the *Session Manager*, is responsible for pushing notifications to the user regarding available time slots, fitness buddies, etc., as discussed in the following subsections. It also allows the user to set her own schedule manually. The system connects to each user's address book and/or social network, and also interfaces with the user's calendar through the *Calendar Integration Manager*.

3.2 Activity Tracking and Classification Module

The system collects accelerometer data from the user's wearable device or phone and feeds them into a classifier that has been trained to recognize and classify the user's activity into one of pre-defined types of activities (e.g. walking, running, jogging, cycling, climbing/descending stairs etc.).

The collected accelerometer data consists of the user id, a timestamp, and the x, y and z values in the accelerometer coordinate system. The activity data needs to be first preprocessed to identify features within a specific time window that will be used as input to the classification process. The time windows are parameters of the system and can be different for each user and each activity. Figure 2 details the process of defining these windows. In the first step, the records in the original data store are grouped by user id and activity. After grouping is done, the records are sorted by timestamp in ascending order. The next step is to identify the so-called "jumps", that are time periods when no

data is collected (the jump time interval is a parameter of the framework that is determined experimentally).

Using the preprocessed accelerometer data, we identify the following features, that are subsequently used as input to the machine learning algorithm that classifies the user's activity in one of the pre-defined types: *average acceleration* (calculated for each axis), *standard deviation* (calculated for each axis), *average absolute difference* (the average of the difference between the value of each input sample records and mean of the total input sample records, calculated for each axis), *average resultant acceleration* (the average of the square root of sum the squares of values of each axis), and *time between peaks* (the time in milliseconds between the peaks in the sine wave for each axis).

The features are calculated for a particular window size such that the user's activity is classified for each of these windows. The feature "*time between peaks*" is useful to find repetitive patterns such as walking and jogging. In our previous work we performed an experimental evaluation of various machine learning algorithms and concluded that tree-based models, and specifically Gradient Boosted Trees, outperformed other methods in correctly classifying the users' activity [6].

Every time the user works out, the *Classification Module* automatically identifies and logs their activity and updates the user's profile. Moreover, the updated profile data is fed in the *Activity Recommendation Engine* that, when applicable, recommends new activities to the user (the *Activity Recommendation Engine* is part of the *Social Recommendation Engine* discussed in Section 4).

3.3 Calendar Integration Manager and Fitness Session Recommendations

The Calendar Integration Manager is responsible for the integration of the user's external calendar accounts with the PRO-Fit application. The PRO-Fit application syncs all the events from all integrated calendars and recommends the best time for the workout as per the goal set by the user. The user can add and delete calendar accounts into the PRO-Fit application through the UI. Credentials are stored securely in PRO-Fit's data store to access the user's calendar data offline. The user's availability and blocked time slots from all integrated calendars, along with the user's profile and activity data are used as input to recommend both fitness sessions, described here, as well as social recommendations, described in Section 4.

As previously mentioned, the user is able to set *weekly or monthly fitness goals* (e.g. total calories to burn, total duration of activities, etc.). Each week, PRO-Fit takes as input the user's *BMI* (Body Mass Index), their *goal*, and their *activity preferences* (originally input during registration and continuously updated through the *Activity Tracker & Classification module*) and recommends several fitness sessions to the user. To achieve this, the system first calculates the time needed to complete each preferred activity, and then finds available slots in their calendar. The user can accept or decline the recommendations. The accepted sessions are registered on the user's calendar, and prior to the beginning of the session, the application sends a push notification on the user's device.

4 Social Recommendation System

The social recommendation system includes two main modules, namely the *Activity Recommendation Engine* and the *Fitness Buddies Recommendation Engine*. The main input to the recommendation process is the $m \times n$ utility matrix of m users and n activities. We denote a user profile as:

$$u = <p_{u1}, ..., p_{un}>, p_{uj} \in [0, r]$$

where p_{uj} represents the preference of user u for activity a_j, r is the maximum rating, and $p_{uj} = 0$ means that the user has not expressed any positive or negative opinion for a particular activity. One major problem for recommender systems, which are based on user profile information is the cold-start problem that refers to new users who have not yet used the system [23]. Since the system does not contain any information about such users activities (and as a result their user profile is empty), it is not possible to provide recommendations based on users with similar profiles. In order to avoid this cold-start problem, the PRO-Fit system requests from new users to rate their favorite activities on a scale 1 to 5 (upon registration) and this information is used to initialize their profiles.

User preference may change over time, and these stated preferences may also differ from the actual activities the user ends up performing. In PRO-Fit the user profiles are not static and the preference scores are updated each time the system logs a user activity. More specifically, the preference score p_{uj} of user u for an activity j is updated using the following formula:

$$p_{uj} = max(p_{uj}, p_{uj} + \frac{|a_{uj}|}{\sum_{j=1}^{n} |a_{uj}|}) \qquad (1)$$

where $|a_{uj}|$ represents the number of times user u performed activity j so far. In essence, the user-provided
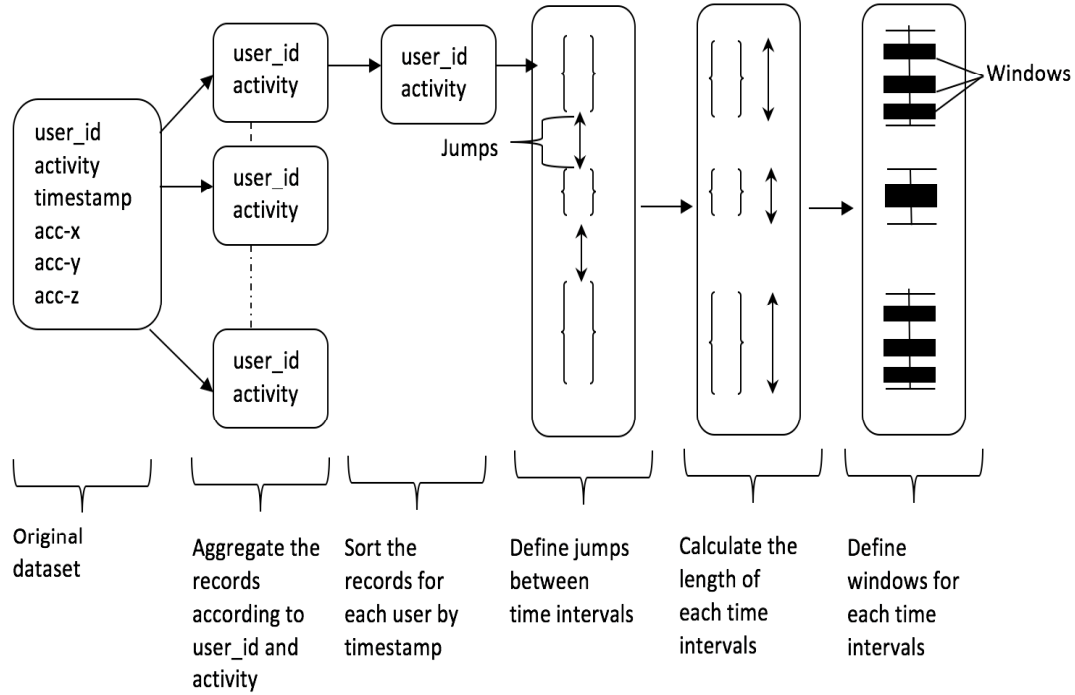
Fig. 2: Data processing architecture

ratings remain unchanged until the user starts participating in that activity. The frequency of participation affects the overall rating thereafter. According to Equation 1, each time the user repeats the activity (i.e. $|a_{uj}|$), the respective preference score increases by a fraction which depends on the total number of performed activities of any kind in the same period (i.e. $\sum_{j=1}^{n} |a_{uj}|$). In order to avoid preference scores to reach the highest value after some time, we normalize the preference scores for all activities at the end of each period, according to the maximum preference score for that period ($p_{uj} = \frac{p_{uj}}{max_{p_{uj}}}$). A month-sized period is used in our prototype implementation but any other period can be used instead. As a result, the preference score for activities that are not performed frequently decreases over time.

4.1 Fitness Buddies' Recommendations

Once the user profiles are established, we calculate the user similarities, based on the intuition that the more

similar likes/dislikes two users have in terms of activities, the more similar they are. For instance, in the toy example depicted in Table 1, we can (empirically) infer that users A and C are more similar than any of them is with B.

To formally calculate such similarities we use the Pearson's correlation coefficient (r) :

$$r(u,v) = \frac{\sum\limits_{i \in I_u \bigcap I_v} (p_{ui} - \bar{p}_u)(p_{vi} - \bar{p}_v)}{\sqrt{\sum\limits_{i \in I_u \bigcap I_v} (p_{ui} - \bar{p}_u)^2} \sqrt{\sum\limits_{i \in I_u \bigcap I_v} (p_{vi} - \bar{p}_v)^2}}$$

$$(2)$$

where I_u denotes the list of activities the user u has performed or rated, $\bar{p}_u$ and $\bar{p}_v$ are the average ratings of users u or v respectively and p_{ui}, p_{vi} are the ratings of users u and v respectively for activity i[1]. The values of r range in the [-1,1], so we normalize them to [0,1] with the following transformation (Equation 3).

---

[1] Actually, Pearson correlation coefficient examines all pairs of ratings assuming 0 when a rating is missing. In order to avoid the effect of missing ratings, we consider only activities rated by both users.

Table 1: Example: User profiles for three users and five activities (preference scale 1-5).

Users/Activities	Walking	Running	Cycling	Cardio	Yoga
A	3	2			5
B		5	4	4	1
C	3	3		5	4

Table 2: Example: User pairwise similarity scores using Pearson's Correlation Coefficient on the matching activities and a normalization to [0,1].

sim(u,v)	A	B	C
A	1.000	0.005	0.854
B	0.005	1.000	0.371
C	0.854	0.371	1.000

$$sim(u, v) = \frac{r(u, v) + 1}{2} \qquad (3)$$

The similarity matrix generated by the data of Table 1 is shown in Table 2.

These scores may be used directly to rank the users based on similarity. In this example, as empirically inferred, user C is more similar to user A than user B. Accordingly, users A and B have a relatively small similarity. However, this is not the final ranked list provided to the user. Instead, we calculate the social similarity, which factors in the strength of the connection between two users of the framework and is defined as:

$$socSim(u, v) = \frac{1}{d[u, v]} \cdot sim(u, v) \qquad (4)$$

where $d[u, v]$ measures the degree of connectivity between users u and v within a social network. This may be implemented in various ways, the simplest being to define the distance between edges u and v in the social graph (so for direct connections, $socSim(u, v)$ is equal to $sim(u, v)$), and is set to a very small number or zero for users with no connection in the social graph. However, more socially-enhanced metrics may be used, such as incorporating the total number of common friends (i.e. the overlap of the two users' social graphs), etc.

Once the social similarities are calculated, the last step is to rank the users based on their geolocation and availability, as the system needs to recommend fitness buddies who are physically located close to the user and available around the same time for exercise. To perform this, the system calculates a mileage radius around each user and rejects users falling outside the radius, while ranking higher the users who are closer to the center of the circle (i.e. the user's current location, as identified by their device's GPS). The *Calendar Integration Module* then helps to filter out the users with conflicting schedules, and the system finally recommends as fitness

buddies those users who are similar in terms of activity preferences and social relations, are close-by, and have non-conflicting schedules.

4.2 Activity Recommendations

In order to recommend new activities, we can employ any collaborative filtering approach, having as input the user-activity matrix introduced previously. In this work, we considered the three most widely used collaborative filtering techniques, namely user-based [1] and item-based [27] collaborative filtering, and matrix factorization [21]. Given a user-activity matrix as input (like the one shown in Table 1, or the example shown in Figure 3), the objective of collaborative filtering is to predict the missing values, which correspond to the missing activity preferences of each user's profile. The intuition behind this prediction process is to find activities of potential interest to the user, that the user has not yet rated (or performed) but other, similar users, have already performed and liked. Once the missing values are predicted, the system ranks and recommends to the user the activities that receive the highest scores. The social network information is employed to re-evaluate the similarity between user preferences. For example, in Figure 3, the preferences of u_1 will be affected by u_2 who is a direct friend but also by u_5 who is a friend of a friend (foaf) and has similar interests in the activities.

In the case of user-based recommendations, the algorithm chooses the top-k users with similar interests to the current user and predicts the preference scores for any activity that the current user has not yet practiced or rated, by averaging the scores of other users. For example, in Figure 3, user u_1 and user u_5 have very similar preferences, based on the ratings they provided, with the $sim(u_1, u_5)$ being at 0.92. So if u_5 has high preference in another activity that is currently missing from u_1's profile, the user-based algorithm will put it high in the list of recommendations. In the case of item-based recommendations, which are more applicable to systems with many users and few items, the algorithm chooses the activities that have a similar preference profile among users with the top ranked activities of the current user and adds them to the recommendation list. For example, in Figure 3, activities 2 and 5 present
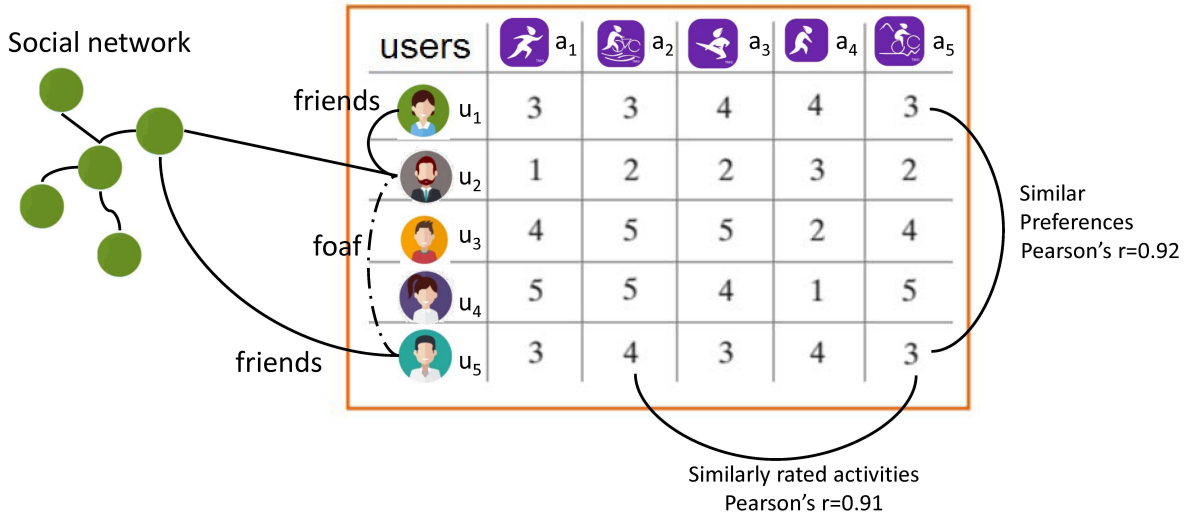
Fig. 3: An illustrative example, comprising social network information and activity ratings

a high similarity in user ratings (Pearson's r is 0.91), which makes either of them an ideal (item-based) recommendation for users who perform the other. Finally, matrix factorization techniques, which scale better in large preference datasets, do not calculate similarities to predict ratings. Instead, taking the same user-activity preference matrix as input, they train a model which learns the latent factors (also called latent features) for users and activities from the actual ratings. The number of factors f is usually smaller than the number of users or activities and matrix factorization predicts the missing preference scores (ratings) by multiplying the user factors by activity factors.

Due to the size of the input data (which in the social recommender context include a social network), and after a preliminary experimental evaluation of the three approaches with small-scale datasets, we decided to adopt a latent factor approach to this problem. In the context of rating-based recommender systems, the intuition behind latent factor models is that both users and items can be characterized by a number of factors. These factors, not necessarily understandable by humans, are inferred by the users' rating patterns. Latent factors can be interpreted in our case as the amount of skills (speed, power, agility etc) required by an activity or the amount of skills that a user is interested in. For example, in Figure 3, the latent factors may correspond to the activity using a bicycle (factor #1) or being a running (factor #2) or static (factor #3) activity. In the new space of latent factors activities 2 and 5, which both include a bicycle (mountain or street) will have high values in the first factor and lower in the other two, whereas the yoga activity (in the third column) will be mapped to high values in the static activity fac-

tor and lower or zero values to the other two factors. Similar mappings will be applied to user preference vectors, through factorization.

The most successful realization of latent factor models for recommender systems is based on matrix factorization. In this approach, the utility (user-activity) matrix is decomposed into a user-factor and an activity-factor matrix. High correspondence between user and activity factors leads to higher predicted ratings.

More formally, let the users U and the activities A be mapped to a joint latent factor space of dimensionality f, where each user u is associated with a vector $x_u \in R^f$ and each activity a is associated with a vector $y_a \in R^f$. In this representation, the elements of x_u and y_a respectively represent the extent to which each user/activity possesses these factors (i.e. the skills that are required or trained by the activity). Therefore, the dot product of the two vectors $y_a^T \cdot x_u$ is expected to capture the interest of the user for the activities' factors, and thus serve to estimate $\hat{p}_{ua}$ as follows:

$$\hat{p}_{ua} = y_a^T \cdot x_u \qquad (5)$$

The training phase of the model, assumes the prediction of several preference scores (similar to $\hat{p}_{ua}$) and compares them to the actual scores given by users (e.g. to the actual p_{ua}). The calibration of the two feature vectors aims at the minimization the regularized square error on the existing user preferences:

$$min_{x,y} \sum (p_{ua} - y_a^T \cdot x_u) + \lambda(\|x_u\|^2 + \|y_a\|^2) \qquad (6)$$

where the constant λ is used to control the extent of regularization. We solve Equation 6 using alternating

Least Squares (ALS) [21]. In essence, ALS alternates between keeping either the x_u's or the y_a's fixed, and recomputes the other by solving a least squares problem. In that way, the factor vectors can be computed independently of each other, resulting in faster real-time responses. Another advantage of ALS compared to other matrix factorization techniques, such as stochastic gradient descent is that it is highly parallelizable with implementations on parallel and distributed platforms such as Mahout or Spark. PRO-Fit employs the ALS implementation of Mahout (ALSWR), an iterative algorithm which aims at minimizing a cost function at each step and stops when a certain stopping criterion is met or a predefined number of iterations (maxIteration) is executed.

Once $\hat{p}_{ua}$ is calculated for each user u, and their non-previously rated activities a, the system ranks all activities and recommends to the user the activities that receive the highest predicted preference score.

4.3 Evaluation of Recommendations

The last but most important step when developing a recommender system is to measure its performance on the basis of how good it predicts user ratings. For this purpose, popular error-based metrics can be employed, such as mean absolute error (MAE) or root-mean-square error (RMSE), which only consider predictions on items in the test set. Error-based metrics are useful under the assumption that a system that provides more accurate predictions is preferred by the user [28].

Assuming that the recommender system generates predicted ratings $\hat{p}_{ua}$ for a test set T of user-activity pairs (u,a) for which the true ratings p_{ua} are known. *Root Mean Squared Error (RMSE)* between the predicted and actual ratings is given by:

$$RMSE = \sqrt{\frac{1}{n} \sum_{(u,a) \in T} (\hat{p}_{ua} - p_{ua})^2} \qquad (7)$$

where n is the size of set T. *Mean Absolute Error (MAE)* is a simpler alternative, given by:

$$MAE = \frac{1}{n} \sum_{(u,a) \in T} |\hat{p}_{ua} - p_{ua}| \qquad (8)$$

5 Experimental Evaluation

Our objective in this experimental evaluation is two-fold: a) to evaluate the activity recommendation engine in terms of prediction accuracy, and b) to assess whether the introduction of the social element improves the overall recommendation process. Unfortunately, while some real-life datasets that include information on users and their preferred activities exist, these do not incorporate any social information. We therefore decided to evaluate the accuracy of the activity recommendation engine using the activity dataset, but without the social input, and employ a non-activity dataset that includes a social graph and can be directly applied to our problem, in order to evaluate our social recommender system. The first experiment aims to evaluate the quality of activity recommendations in the absence of social information, using real activity data for building and evaluating the activity preference model. The second experiment aims to highlight that the use of social information can be beneficial for the quality of recommendations. We discuss our findings in the subsections that follow.

5.1 Evaluation of Activity Recommendations

In order to evaluate how accurately the system is able to predict the preferred activities of each user, we used the Actitracker dataset collected by Kwapisz et al. [22], publicly available through the wireless sensor data mining (WISDM) lab[2] [25]. This dataset includes activity data collected from android devices of a number of volunteers who participated in the experiment performing one or more out of 6 distinct activities. Although the tracked activities are not strictly related to fitness (for example standing or lying down) they do expose each user's preferences of specific physical activities versus others, and are recorded in a level of detail (in terms of data collected) that reflects what would be collected by the PRO-Fit application. Since this real-life dataset has been used previously to evaluate similar systems, we decided to use it in our evaluation as well. The data was collected at an interval of 50ms, which means it contains 20 samples per second. It contains user information along with their tri-axial axis information and the timestamp showing when it was collected. Since our aim is to evaluate the activity recommendation algorithm and not the activity classification algorithm[3], we employed only the manually labeled dataset, which comprises data from 225 users and kept only the first three attributes of each tuple, namely the *user_id*, *activity* and *timestamp*. In order to calculate the preferences of a user for a particular activity, we calculated the normalized sum of each activity for each user, as described in Equation 1 (please note, that in this dataset, the user preferences p_{uj} are initially set to 0 as no such explicit

---

[2] http://www.cis.fordham.edu/wisdm/dataset.php#actitracker
[3] This algorithm was evaluated in our previous work [11].

Table 3: RMSE on validation set for Matrix Factorization
algorithm ($\lambda = 0.1$)

RMSE					
f \\ λ	0.01	0.05	0.1	0.5	1
5	0.1999	0.1925	**0.1868**	0.2411	0.256
10	0.2016	0.1912	0.1877	0.2412	0.256
20	0.1936	0.192	0.1887	0.2412	0.256
30	0.2023	0.1916	0.1889	0.2412	0.256

Table 4: MAE on validation set for Matrix Factorization
algorithm ($\lambda = 0.1$)

MAE					
f \\ λ	0.01	0.05	0.1	0.5	1
5	0.1179	0.1141	**0.1103**	0.1455	0.1593
10	0.1176	0.114	0.1113	0.1454	0.1593
20	0.1243	0.1142	0.1112	0.1454	0.1593
30	0.1295	0.1138	0.1113	0.1454	0.1593

information is provided). As a result, we ended with a dataset comprising 480 (activity) preference scores for the 225 users.

In order to find the optimal values for the parameters λ and f of the Matrix Factorization model, we performed multiple random sampling (5 times), on a 60/20/20 split (for training/validation/testing respectively) measuring the RMSE and MAE errors (both measuring the error between the predicted and the actual value of the activity preference). Looking at the results of the evaluation, shown in Tables 3 and 4, we observe that the optimal values are given when $\lambda = 0.1$ and $f = 5$ for both RMSE and MAE.

At this point we would like to point out that we have performed similar experiments (on the same dataset) for the two most popular neighborhood collaborative filtering algorithms, namely user-based and item-based collaborative filtering. From that experimental evaluation, we observed that user-based collaborative filtering was performing similarly to matrix factorization. Due to the scalability and flexibility advantages of the matrix factorization approach, and its ability to handle better sparse matrices, we decided to adopt this algorithm in the prototype's implementation.

5.2 Evaluation of Social Recommendations

As mentioned previously, there is currently no activity dataset available that also incorporates a social element. Therefore, we had to employ a different dataset, that could be used as an approximation of the users having activity preferences scenario, while incorporating a social element at the same time. Our goal is to evaluate whether the preferences of close friends can affect (positively) the recommendations, when weighed more heavily in the recommendation process. For this reason, we decided to use the well-known and broadly used Yelp challenge dataset[4]. This is a very rich dataset, but for the purposes of our study we only used the following information for each user: *user id, friends ids, restaurants rated, ratings*. Of course, restaurant preferences are not fitness activity preferences. However, they still have an activity preference aspect and, more importantly, such preferences can be affected by close friends. The premise is very similar in both scenarios: in the PRO-Fit recommendations, we expect that users will share similar activity preferences with those who are closer to them (in a social network context), while in the restaurant context, we assume that users will share similar preferences with their closest friends (again reflected as connections in the virtual social network).

Similarly to the previous set of experiments, our main input is the triplet *user id, restaurant* (as in *activity*), *rating* (as in *preference*). However, a fourth very important input is the *social network* of friends and friends-of-friends of the user. Since Yelp's social network is secondary to its main goal (of allowing users to rate businesses), it is rather sparse. Moreover, it only made sense to select user connections who share the same restaurant preferences, i.e. reside in the same area. Therefore, we focused only on the city of Las Vegas, and selected, as starting "seed" the 2030 users who had at least one rating of a Las Vegas business and the highest number of friends. We then performed five experiments, using different instantiations of the social similarity metric $socSim(u,v)$ of Equation 3, depending on the value of the social connectivity parameter $d[u,v]$, that in turn resulted in a different input dataset:

Experiment 1 1-degree of separation (OneD): In this experiment we expanded the original "seed" dataset of 2030 users with their connections. In this experiment, $d[u,v] = 1$ when users u and v have one degree of separations and 0 otherwise. Let F_u denote the "friends" of user u (i.e. u's direct connections in the social graph), then Equation 4 becomes:

$$socSim(u,v) = \begin{cases} sim(u,v), & \text{if } v \in F_u \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Experiment 2 Baseline One (BasOneD): In this experiment we expanded the original "seed" dataset of 2030 users to become as big (in terms of number of users) as that of the one in Experiment 1, by randomly selecting the additional users. Let B_A denote the set of users that were randomly selected in this experiment,

---

[4] https://www.yelp.com/dataset_challenge

Table 5: Input datasets

Dataset	# Users	#Businesses	#Ratings
OneD	61,128	16,009	435,129
BasOneD	61,128	11,105	235,869
TwoD	108,191	16,934	1,003,766
BasTwoD	108,191	11,738	407,206
BasAll	146,763	17,386	1,022,826

social similarity is computed as:

$$socSim(u,v) = \begin{cases} sim(u,v), & \text{if } v \in B_A \\ 0, & otherwise \end{cases} \quad (10)$$

Experiment 3 2-degrees of separation (TwoD):
In this experiment we expanded the original seed dataset of 2030 users with their friends and friends of their friends. In other words, we expanded the social graph of each user to include all users within two degrees of separation. In this experiment, we set $d[u,v] = 1$ when users u and v have one degree of separation, $d[u,v] = 1/2$ when users u and v have two degrees of separation, and 0 otherwise. Let F_u denote the friends of user u, then Equation 4 becomes:

$$socSim(u,v) = \begin{cases} sim(u,v), & \text{if } v \in F_u \\ \frac{sim(u,v)}{2}, & \text{if } v \in F_z \& z \in F_u \\ 0, & otherwise \end{cases} \quad (11)$$

Experiment 4 Baseline Two (BasTwoD): In this experiment we expanded the original "seed" dataset of 2030 users u to become as big (in terms of number of users) as that of the one in Experiment 3, by randomly selecting the additional users. Let B_B denote the set of users that were randomly selected in this experiment, then social similarity is computed as:

$$socSim(u,v) = \begin{cases} sim(u,v), & \text{if } v \in B_B \\ 0, & otherwise \end{cases} \quad (12)$$

Experiment 5 Baseline C (BasAll): In this experiment we used the entire Las Vegas dataset as input to the recommendation algorithm, eliminating the social factor.

We should point out that, since the users used as input in each experiment are different, the number of ratings and businesses are also different. The numbers of users/businesses/ratings for each of the five experiments are shown in Table 5.

We calculated the RMSE and MAE using a 60/20/20 split and 10-fold cross-validation for all possible combinations of the following parameters: $f = \{5, 10, 20\}$, $\lambda = \{0.01, 0.05, 0.2, 0.5, 1\}$, and *number of iterations*

Table 6: RMSE and optimal parameter settings for various input datasets

Experiment	f	λ	numIter	RMSE
#1 OneD	20	0.50	20	1.215
#2 - BasOneD	30	0.50	20	1.27
#3 - TwoD	30	0.10	20	**1.09**
#4 - BasTwoD	5	0.50	20	1.28
#5 - BasAll	20	0.50	10	1.18

Table 7: MAE and optimal parameter settings for various input datasets

Experiment	f	λ	numIter	RMSE
#1 OneD	30	0.50	20	0.954
#2 - BasOneD	10	0.50	10	0.917
#3 - TwoD	30	0.05	20	0.782
#4 - BasTwoD	10	0.50	20	0.925
#5 - BasAll	30	0.50	20	0.821

$= \{10, 20\}^5$ for the five input datasets. Tables 6 and 7 show the combinations that generated the best (lowest) RMSE and MAE errors for each experiment. We observe that in both cases, the social recommender system that uses as input the extended network spanning two degrees of separation for each "seed" user, outperforms the remaining approaches. This verifies our intuition that incorporating the social element results in more accurate preference predictions. Although there is no direct comparison between the results of this and the previous experiment that evaluated the activity recommendations (Section 5.1), the increased MAE and RMSE values in the second experiment can also be explained by the different scale of user ratings in the two datasets. More specifically, the user ratings in the Yelp dataset have been normalized to a 0-10 integer scale, since the original dataset contained 0-5 'star' ratings allowing for half stars, whereas the ratings we created for user activity preferences were on a 0-5 integer scale. As a result, MAE values of 0.1 on a 0-5 scale rating and of 1 on a 0-10 scale are considered very good.

6 User study

In order to evaluate the usability of the PRO-Fit application and measure the engagement of end users, we conducted a user study including 24 people, who were relatively active (workout for at least 3 days a week or

---

[5] The (maximum) number of iterations is one of the stopping criteria for the iterative implementation of ALS for Mahout that we employed. The more iterations, the most possible for the algorithm to find an optimum solution. Using 15 iterations is a reasonable default to try (https://mahout.apache.org/users/recommender/intro-als-hadoop.html). However, we also tested 10 and 20 iterations

more). The user group consisted of 16 males and 8 females, between 20 and 35 years old, all residing in the broader San Francisco bay area. We requested the users to install the PRO-Fit app on their smartphones and asked them to use it for 15 days[6]. At the end of the 15-day period, we asked the participants to respond to a set of questions and also give us some verbal feedback. Table 8 includes a summary of the survey questions and responses. Below are some of the comments received from the survey:

— "I really liked the session reschedule recommendation as it automatically detected conflicting sessions"
— "I like the app feature of scanning all my calendar events and providing proactive session recommendations without any manual pain to organize things"
— "Overall, the app experience and its functionality is good, however there can be lot of improvements in the performance of the app."
— "I found this app really amazing to plan out my exercise plan for over a 2-week period, which really motivated me to strictly follow the plan until completed."
— "Looks interesting, but should reduce the data consumption"
— "Great app, useful for lazy and laid back people to get motivated and workout :) "

From the quantitative and qualitative responses we observed that even in its prototype mode, the PRO-Fit application was successful in motivating the majority of the participants, with automatic session scheduling and rescheduling whenever they added entries to their personal/work calendars, being the most popular features. Some of the criticism we received had to do with the performance of the application itself, something expected as this application still runs in prototype mode. We should also note that the socially enhanced part of the application could not be thoroughly evaluated in this small-scale user study as there was no overlap between the participants' contacts and the rest user group. Finally, we observed that the majority of users said that this application was successful in motivating them to keep their planned activities, and that they would recommend it to a friend.

7 Conclusions

In this paper we presented PRO-Fit, a personalized fitness assistant framework that integrates activity data collected by the user's wearable device or smart phone,

their activity preferences and fitness goals, their availability and their social network, and automatically generates fitness schedules and socially-enhanced recommendations of new activities, as well as fitness buddies. One of the issues faced by this study is that there does not exist a publicly available dataset that comprises both social information and activity related preferences. For this reason, we used two different datasets: one for extracting physical activity preferences and one that includes both user preferences and social network information collected in a different context. Our experimental evaluation, including two real-life datasets, showed that the socially-enhanced recommendations, outperform the non-socially enhanced ones. More specifically, the algorithm that predicts a user's preference when the social network includes connections up to two degrees of separation, was the most accurate among all we examined.

In the evaluation, we experimented with three different parameters of the matrix factorization algorithm for collaborative filtering and found that the optimum number of latent factors was 5. However, one must be careful in a different setup. For example, if the system detects 60 activities instead of 6, or monitors activities with completely diverse skills' profile, then the optimum number of latent factors may differ. In the second experiment, the structure of the social network (e.g. if it is more dense or sparse), will affect the choice of parameters. However, the experiments with several parameter combinations reported the best results when the second degree friends (friends of a friend) were used for recommending activities.

We also performed a user study having 24 participants use the PRO-Fit prototype to set fitness goals and schedule activity sessions. As with any pilot study, there are limitations to the validity of any resultant claims made. However, the results from this study were very encouraging and, even though the participants used a prototype, the majority of them rated it very positively. A detailed analysis of usage analytics that will highlight the time of the day they tend to accept recommendations, their preference to social-based or similarity-based recommendations etc, is within our next steps and is expected to improve the system performance in the user acceptance tests.

As part of our future work, we plan to explore and evaluate different ways to calculate the tie strengths (and as a consequence the social similarity) between the users of the application, and extend the prototype to incorporate external social networks in addition to the user's contacts, such that we can better evaluate empirically the user's experience. Moreover, we aim to publicize anonymous data from the PRO-Fit social net-

---

[6] A more detailed description of the PRO-Fit prototype as well as application screenshots can be found in [12]

Table 8: PRO-Fit user study responses' summary

Question		Responses (Range or %)
How many sessions did you create manually per week?		2 to 6
How many session recommendations did you get per week?		4 to 7
How many calendar events do you have on average per day?		0 to 5
Did you find session rescheduling useful	Yes	60%
	No	40%
How many reschedule recommendations did you accept?		58% Accepted 42% Rejected
What time did you prefer to schedule your session?	Morning	30%
	Afternoon	15%
	Evening	55%
Do you feel the PRO-Fit app helped you be motivated to do your workout?	Yes	45%
	No	55%
Do you feel the PRO-Fit app helped you be motivated to do workout on scheduled time?	Yes	70%
	No	30%
How do you rate the rescheduling sessions placed?		Average 60% useful
Did your analytics help you to know your workout pattern?	Yes	67%
	No	33%
What is the best feature you like about the app you are using?	Session recommendation	25%
	Session reschedule	45%
	Goal tracking	30%
Would you like to recommend PRO-Fit app to your friends & family?	Yes	64%
	No	36%

work and activity preference database, once we reach a reasonably sized user-base.

References

1. G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.
2. A. Ahtinen, M. Isomursu, M. Mukhtar, J. Mäntyjärvi, J. Häkkilä, and J. Blom. Designing social features for mobile and ubiquitous wellness applications. In *Proceedings of the 8th International Conference on Mobile and Ubiquitous Multimedia*, page 12. ACM, 2009.
3. I. Anderson, J. Maitland, S. Sherwood, L. Barkhuus, M. Chalmers, M. Hall, B. Brown, and H. Muller. Shakra: tracking and sharing daily activity levels with unaugmented mobile phones. *Mobile networks and applications*, 12(2-3):185–199, 2007.
4. A. Anjum and M. Ilyas. Activity recognition using smartphone sensors. In *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, pages 914–919, 2013.
5. M. A. Ayu, T. Mantoro, A. F. A. Matin, and S. S. Basamh. Recognizing user activity based on accelerometer data from a mobile phone. In *Computers & Informatics (ISCI), 2011 IEEE Symposium on*, pages 617–621. IEEE, 2011.
6. A. Bajpai, V. Jilla, V. N. Tiwari, S. M. Venkatesan, and R. Narayanan. Quantifiable fitness tracking using wearable devices. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 1633–1637. IEEE, 2015.
7. D. Bernardes, M. Diaby, R. Fournier, F. FogelmanSoulié, and E. Viennet. A social formalism and survey for recommender systems. *ACM SIGKDD Explorations Newsletter*, 16(2):20–37, 2015.
8. D. Centola. An experimental study of homophily in the adoption of health behavior. *Science*, 334(6060):1269–1272, 2011.
9. Y. Chen and P. Pu. Healthytogether: exploring social incentives for mobile fitness applications. In *Proceedings of the Second International Symposium of Chinese CHI*, pages 25–34. ACM, 2014.
10. J. Comstock. Mobihealthnews: Half of mobile health app users are using fitness apps. http://mobihealthnews.com/30199/half-of-mobile-health-app-users-are-using-fitness-apps/ [Online; posted 21-February-2014].
11. S. Dharia, V. Jain, J. Patel, J. Vora, S. Chawla, and M. Eirinaki. Pro-fit: A personalized fitness assistant framework. In *Proc. of the 28th Intl. Conf. on Software Engineering and Knowledge Engineering (SEKE 2016), Redwood City, CA*, 2016.

12. S. Dharia, V. Jain, J. Patel, J. Vora, R. Yamauchi, M. Eirinaki, and I. Varlamis. Pro-fit: Exercise with friends. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*, pages 1430–1433. IEEE, 2016.

13. R. Dishman and J. Buckworth. Increasing physical activity: a quantitative synthesis. *Medicine and science in sports and exercise*, 28(6):706, 1996.

14. D. L. Feltz, N. L. Kerr, and B. C. Irwin. Buddy up: the köhler effect applied to health games. *Journal of Sport and Exercise Psychology*, 33(4):506–526, 2011.

15. S. Ghose and J. J. Barua. A systematic approach with data mining for analyzing physical activity for an activity recognition system. In *Advances in Electrical Engineering (ICAEE), 2013 International Conference on*, pages 415–420. IEEE, 2013.

16. J. J. Guiry, P. van de Ven, and J. Nelson. Classification techniques for smartphone based activity detection. In *Cybernetic Intelligent Systems (CIS), 2012 IEEE 11th International Conference on*, pages 154–158. IEEE, 2012.

17. N. Gupta and S. Jilla. Digital fitness connector: Smart wearable system. In *Informatics and Computational Intelligence (ICI), 2011 First International Conference on*, pages 118–121. IEEE, 2011.

18. I. Guy. Social recommender systems. In *Recommender Systems Handbook*, pages 511–543. Springer, 2015.

19. S. Halko and J. A. Kientz. Personality and persuasive technology: An exploratory study on health-promoting mobile applications. In *International Conference on Persuasive Technology*, pages 150–161. Springer, 2010.

20. J.-H. Hong, J. Ramos, and A. K. Dey. Toward personalized activity recognition systems with a semipopulation approach. *IEEE Transactions on Human-Machine Systems*, 46(1):101–112, 2016.

21. Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

22. J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

23. X. N. Lam, T. Vu, T. D. Le, and A. D. Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 208–211. ACM, 2008.

24. Y. Li and Y. Zhang. Application of data mining techniques in sports training. In *Biomedical Engineering and Informatics (BMEI), 2012 5th International Conference on*, pages 954–958. IEEE, 2012.

25. J. W. Lockhart, G. M. Weiss, J. C. Xue, S. T. Gallagher, A. B. Grosner, and T. T. Pulickal. Design considerations for the wisdm smart phone-based sensor mining architecture. In *Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data*, pages 25–33. ACM, 2011.

26. L. A. Messé, G. Hertel, N. L. Kerr, R. B. Lount Jr, and E. S. Park. Knowledge of partner's ability as a moderator of group motivation gains: An exploration of the köhler discrepancy effect. *Journal of Personality and social Psychology*, 82(6):935, 2002.

27. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

28. G. Shani and A. Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011.

29. P. A. Silva, K. Holden, and A. Nii. Smartphones, smart seniors, but not-so-smart apps: A heuristic evaluation of fitness apps. In *International Conference on Augmented Cognition*, pages 347–358. Springer, 2014.

30. J. Tang, X. Hu, and H. Liu. Social recommendation: a review. *Social Network Analysis and Mining*, 3(4):1113–1133, 2013.

31. M. T. Uddin and M. A. Uddiny. Human activity recognition from wearable sensors using extremely randomized trees. In *Electrical Engineering and Information Communication Technology (ICEEICT), 2015 International Conference on*, pages 1–6. IEEE, 2015.

32. F. M. F. Wong, Z. Liu, and M. Chiang. On the efficiency of social recommender networks. *IEEE/ACM Transactions on Networking*, 24(4):2512–2524, 2016.

33. Y. Zhang, M. Chen, D. Huang, D. Wu, and Y. Li. idoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization. *Future Generation Computer Systems*, 66:30–35, 2017.