SocialFusion: Context-Aware Inference and Recommendation By Fusing Mobile, Sensor, and Social Data

Aaron Beach[1], Mike Gartrell[1], Xinyu Xing[1], Richard Han[1], Qin Lv[1], Shivakant Mishra[1], Karim Seada[2]

[1]University of Colorado at Boulder, [2]Nokia Research Center Palo Alto

Contact: {aaron.beach, mike.gartrell, xinyu.xing}@colorado.edu

Department of Computer Science
University of Colorado at Boulder

Technical Report CU-CS-1059-09

December 2009

SocialFusion: Context-Aware Inference and Recommendation By Fusing Mobile, Sensor, and Social Data

Aaron Beach[1], Mike Gartrell[1], Xinyu Xing[1], Richard Han[1], Qin Lv[1], Shivakant Mishra[1], Karim Seada[2]
[1]University of Colorado at Boulder, [2]Nokia Research Center Palo Alto
{aaron.beach, mike.gartrell, xinyu.xing}@colorado.edu

ABSTRACT

Mobile social networks are rapidly becoming an important new domain showcasing the power of mobile computing systems. These networks combine mobile location information with social networking data to enable fully context-aware environments. This paper proposes SocialFusion, a framework to support context-aware inference and recommendation by fusing together mobile, sensor, and social data. We investigate a case study of SocialFusion that enables an application for group-based context-aware video. We show how SocialFusion can be used to gather mobile, sensor, and social data, infer group descriptors, and then apply these meta-level descriptors to improve the recommendation of video playback for a group of users viewing the same screen.

1. INTRODUCTION

Mobile social networks are in their infancy, but have the potential to revolutionize the field of mobile computing by enabling fully context-aware inference and recommendation in ubiquitous computing environments. To date, a variety of independent research projects have begun to show the power of mobile social networks, e.g. Serendipity [15], WhozThat [8], and CenceMe [29]. We believe it is an opportune time to evolve a more systematic framework upon which researchers can build a new foundation for mobile context-aware computing. This paper presents SocialFusion, a framework for systematically fusing together diverse input streams from mobile smartphones, sensor networks, and social networks so that inferences can be made from the assembled data to extract contextual clues and recommendations can then be made based on these inferences to invoke the appropriate decision(s) and/or action(s).

SocialFusion is motivated by the observation that the phenomenal growth of social networks, mobile smartphones, and sensor networks provides an unparalleled opportunity to achieve a more comprehensive understanding of the context surrounding an individual in nearly any given environment. Mobile devices are nearly ubiquitous and can be leveraged to provide location and announce a user's identity/presence in a room or place [8, 15]. In addition, mobile smartphones have been enriched with a multitude of sensors, such as accelerometers, microphones, cameras, and even digital compasses, that can be mined to infer actions [29] or even orientation. Environmental sensor networks, both indoor and outdoor, provide the ability to monitor light, temperature, humidity, wind, and other parameters that further enrich our understanding of context. Finally, social networks like Facebook, MySpace, Twitter and LinkedIn provide invaluable contextual information concerning the preferences of individuals, e.g. their favorite hobbies, bands, and films, as well as their relationships with one another. Moreover, all three classes of mobile, sensor, and social data when temporally archived provide historical perspectives that further enhance the understanding of context. The confluence of the explosive growth in social networks, sensor networks, and mobile computing provides us with the ideal opportunity to *fuse together* all of these diverse input streams and thereby achieve an optimal understanding of individual and group context, which can be leveraged to create a new frontier of context-aware applications.

Our observation is that it is the *combination* of mobile, sensor, and social data that can help build better context-aware applications, and that each data stream independently only provides an incomplete picture of situational awareness. For example, recent mobile health applications have been deployed to help monitor disease outbreaks, but currently only list reported disease hot spots [3] rather than an individual's exposure risk, i.e. how close they may have come into contact with other affected individuals. By enhancing such applications with detailed location information obtained from an individual's smartphone, the application would be able to more accurately assess whether the individual was exposed to a disease, by comparing to other location traces of individuals known to have contracted the disease. However, the granularity of location provided by mobile location services often only localizes an individual to the scale of a room or lecture hall, which may be insufficient to identify whether two individuals were in close enough proximity to communicate the disease. By integrating additional social networking information, namely friendship information, then the application would have a stronger inference of potential contact, namely two friends are more likely to sit together in the same meeting room and thus more likely to have come into contact. Note that the friendship information alone would not provide enough context absent the location. It is the combination of social networking information with mobile location information that makes a much stronger case that there was the potential risk of disease exposure, and thereby substantially enhances the existing health application with context awareness.

Multimedia audio and video applications can also benefit from the fusion of mobile, sensor, and social data. For example, a music jukebox player has been enhanced to play customized location-aware and preference-aware music, namely it is aware of both the identity of individuals within a room, via their cell phones, as well as the preferences of those individuals in the room, via their social networking preferences, e.g. the list of their favorite bands [8]. Similarly, a video player has been enriched with context so that it plays trailers that are customized to the tastes of people viewing a shared video screen together [19]. Note that in these applications, a new facet of context awareness emerges, namely the research challenge is not only of inferring characteristics about an individual, e.g. their tastes, but even more challenging how to infer meaningful characteristics of a group of individuals. It is the collective tastes of the individuals that would influence the recommendation of a film or song to a group of people.

SocialFusion is designed to enable such applications by providing the framework whereby applications can obtain mobile, sensor, and social data, integrate them, infer context, and decide on a suitable course of action. The challenges addressed by SocialFusion include how to collect diverse data streams, how to protect the privacy of the collected data, how to analyze and fuse together the assembled data to infer higher-level contextual meaning, e.g., the characteristics of a group, and how to use what we have learned from inference to effectuate appropriate actions for different kinds of context-aware applications.

Figure 1 illustrates SocialFusion's multi-stage computing framework. The first stage collects together data from three major classes of data input streams, namely social networks, mobile phones, and sensor networks. The second stage incorporates inference functionality whose task is to fuse the data and thus derive higher-level contextual meaning in the form of descriptors from the raw data. These descriptors, combined with the original data, are then supplied to a third stage consisting of a recommendation engine that decides what kind

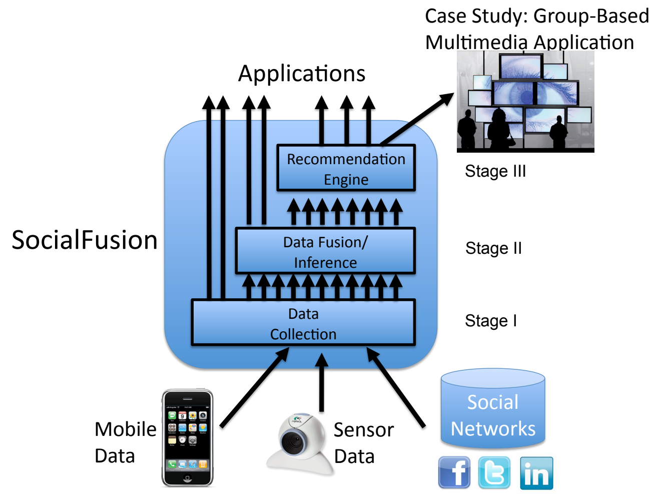


Figure 1: SocialFusion's multi-stage computing framework for inferring context and recommending contextual action by fusing mobile, sensor, and social data.

of context-aware action to take. The contributions of SocialFusion consist of the following:

- introducing the concept of fusing together social networks, mobile smartphones, and sensor networks to improve context awareness, especially involving social situations
- designing a multi-stage platform for multi-dimensional fusion consisting of data collection/management, data inference/fusion, and recommendation stages
- using the platform to implement a particular context-aware application which is a case study involving recommending films to show to a group of people
- showing how the additional context awareness supplied by SocialFusion's diverse input streams improves a context-aware application's actions, i.e., how the addition of social networks and group descriptors improves the group recommendation generated by a context-aware video application

2. RELATED WORK

Prior research has addressed portions of the SocialFusion vision of integrating social networks, mobile phones, and sensor networks, but as far as we are aware, no single project has addressed the full scope proposed here for SocialFusion. In the space of mobile social networks, Serendipity links together a stand-alone social network with mobile devices [15], but is not integrated with existing social networks like Facebook. WhozThat integrates Facebook with mobile phones to provide context-aware audio, but does not integrate with any sensors [8]. CenceMe mines mobile data provided by iPhone sensors to infer user actions, but is not integrated with social networks [28]. Early work on smart spaces occurred largely prior to the advent of social networks [22, 16]. Commercial location-aware mobile social networking services such as Brightkite [1] and Loopt [2] often devise their own social networks, with little to no exploitation of full context awareness as proposed here.

Recommendation algorithms and systems have received significant attention from both academia and industry since the mid-1990s when collaborative filtering was introduced. Recommendation systems are usually classified into two categories: content-based recommendations and collaborative filtering based recommendations. Content-based recommendation systems recommend an item to a user based on item description and user's interests [10, 25] and are useful recommending webpages, news articles, items for sale, etc.

Collaborative filtering based systems recommend items that other similar users have preferred [23, 6]. Collaborative recommendation computes the similarity to other users rather than to other items. Several hybrid recommendation systems combine both collaborative and content-based methods [7, 12].

The majority of the research in recommendation systems has been focused on making recommendations to individual users. Making recommendations to a group of users with potentially competing interests introduces a whole set of new challenges. A few projects that have addressed the problem of making recommendations to a group of users include [18, 30]. SocialFusion provides support for making group recommendations based on all these approaches by exploiting the advantages of content-based, collaborative and hybrid methods. In addition, it enhances the current state-of-the-art by incorporating detailed context information obtained by fusing data from online social networks, mobile smartphones and sensor networks. In particular, this detailed context information aids in better identifying the characteristics of a group as well as the current environmental conditions, which in turn results in making better group recommendations.

An emerging body of work seeks to identify groups from individual data traces by applying graph theory clustering approaches. Standard solutions to the clustering problem include the K-means or Fuzzy C-means algorithms. Convoy detection seeks to identify a group traveling together by using density-based analysis of trajectory databases [21]. Fuzzy clustering has been used as a technique to track robotic objects [24]. Sensors in the environment or on phones, such as accelerometers, have been used to identify groups [34]. Section 5 describes how we build upon each of these techniques.

The design of SocialFusion has focused its privacy concerns on two major issues. The first issue is presence sharing or more generally the association of users with a specific context. The second issue has to do with the release of personal information in general and whether information deemed private by a user can still be useful to the system without compromising that user's privacy.

Previous work has dealt with the issue of anonymous presence sharing between users through matching their shared points in time and space [14, 26, 9]. This is achieved through announcing anonymous identifiers which are resolved through a trusted broker system. This work has formed the basis for anonymous presence sharing in SocialFusion.

The second problem of sharing personal or private information without compromising the privacy of the individual involves guaranteeing the disassociation of all publicly released private data from its associated user. The private data must be "disassociated" in the sense that, under some guarantee, it cannot be re-associated with its private source identity. This problem is closely related to the K-anonymity problem in which a released piece of data is K-anonymous if it maps to no fewer than K different sources. Prior work on K-anonymity in social networking data has largely focused on developing algorithms that anonymize only the social graph of friendships [27, 33, 31], or both friendship and user profile data obtained from social networks [13]. [27, 33, 31] primarily involve perturbation of the social graph structure in a social network. Many of these approaches also are designed for off-line anonymization, whereas SocialFusion must react quickly to the appearance of a user in an environment. Instead, SocialFusion offers a new approach to anonymization that does not allow the perturbation of social links nor the modification or generalization of a user's data, and in addition supports real-time mobile social networking applications. We sketched an approach in [9], but offer the complete solution and evaluation in this paper.

3. SOCIALFUSION FRAMEWORK

Figure 2 provides a detailed example of SocialFusion's multi-stage computing framework. SocialFusion divides its data flow into a sequence of three stages with clearly separated functionality in each stage. The framework is designed to be general, so that new modules can be inserted and built on top of existing modules in a hierarchical fashion. In this case, the individual modules pictured were each developed for our case study of a group-based context-aware video application, and demonstrate how the components of each stage fit together. The modules with solid borders have been completed and integrated into the SocialFusion framework, while the modules with dotted borders/lines have yet to be integrated into the full framework.

The first stage collects preference data from Facebook and Netflix, and location data from mobile phones, and places the information into a set of databases. As there are both location privacy and data privacy concerns

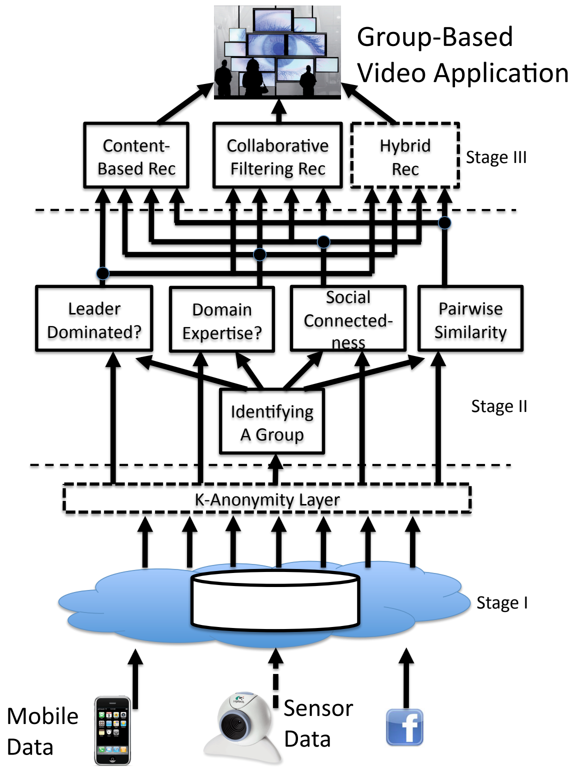


Figure 2: A detailed example of how SocialFusion's multi-stage computing framework can be leveraged t o support a group-based context-aware video application.

with releasing such data, SocialFusion incorporates an anonymization layer in Stage I that manages the release of any of the collected information. Any release of information from its databases must satisfy a K-anonymity criterion. Thus, any subsequent stage may only operate on data that has been sufficiently anonymized. This approach protects each user's privacy as soon as possible after data collection.

Stage II then applies a hierarchical set of *describers* to the assembled data in order to extract or infer contextual meaning. A describer is a module with an inference algorithm that is tailored for analyzing a specific characteristic of the assembled data. A describer module generates a *descriptor* parameter that assesses the degree to which the characteristic is possessed by the analyzed data. For example, we have built a describer module for our case study that analyzes collected individual mobile location traces as well as Facebook friendship and preference information in order to extract groups of individuals that are traveling together, generating a group ID descriptor that identifies the individuals in the group.

As shown, describers can be hierarchically organized in Stage II to base their inferences upon the inferences of other prior describers. For example, each of the four describer modules Leader, Expert, Social, and Similarity depends for their inferences upon the group ID descriptor generated by the group identifier describer, i.e. in order for the Leader describer to analyze whether the social graph of relationships within a group reveals the presence of a dominant social leader, it must first know the membership of the group, which is revealed in the group ID descriptor.

Once the describer modules have inferred sufficient contextual clues from the assembled data in the form of descriptors, then Stage III uses these inferences to recommend a context-aware action. The recommenders will have access to not just the descriptors but also the collected raw data in the database via the anonymity layer, though this is not shown in the figure for the sake of clarity. For example, a set of describers may ideally determine individual users' current activity or task, their emotional mood, their physical health, and their musical tastes. These descriptors can be used as input to a recommendation algorithm that decides what kind of music to play to a user, e.g. knowing a user is exercising and likes rock music, then the recommender chooses an up-tempo rock tune to play for the individual.

We expect that as data gathering, inference modules, and recommendation algorithms become more sophisticated that contextual detail will become ever more enriched and the power of SocialFusion to transform our environments and spaces with adaptive context awareness will be even more starkly demonstrated. As a result, we have focused our initial investigation of SocialFusion on a case study that involves rich contextual detail, namely a group of individuals, who not only have their individual preferences, but also sophisticated social relationships to one another rich with contextual detail that affect what kind of recommendation should best be made. Focusing on context-rich group behavior gives SocialFusion the best opportunity to demonstrate its utility towards improving the customization of context-aware applications. Moreover, since groups represent a common mode of human interaction, then SocialFusion will be shown to address a common case. However, focusing on group behavior also poses a greater challenge in that determining a recommendation for a group of individuals is inherently more difficult than determining a recommendation for a single individual, which is already a difficult problem in and of itself.

Our SocialFusion project makes headway in solving the group-based context-awareness problem, and uses a group-based video application to do so, as shown in Figure 1. The group descriptors generated by the Stage II describers are then used as input in Stage III to the set of recommendation algorithms investigated in this case study, namely content-based recommendation algorithms, collaborative filtering recommendation algorithms, and a hybrid of the two. These recommendations are compared in our evaluation section against ground truth obtained from our user studies of groups.

4. COLLECTING AND MANAGING DATA

The power of SocialFusion comes from the combination of data obtained from a variety of sources. SocialFusion organizes input data streams into three major classes, namely mobile data from smartphones, sensor data from fixed sensor networks, and social networking data from online social networks. In some cases, SocialFusion requires support from the sources from which these data streams are generated, e.g. by installing a mobile application on smartphones. In other cases, SocialFusion uses the standard APIs provided by the data source to collect the relevant as and when needed.

Mobile data from smartphones provides important contextual information of users as well as groups of users as they move from one place to another. This includes location data, accelerometer data, digital compass data, as well as pictures and videos captured using phone's camera. This data is collected by installing a mobile application on the smartphones of the users. The application retrieves a user's location and other sensor values and passes on that information to SocialFusion. Since this information is time dependent and varies as users move, the mobile application passes on fresh values at regular intervals. Our application allows the user to adjust the tradeoff between update frequency and battery power.

Data from fixed sensors embedded in the local smart space, e.g. temperature, humidity, and infrared sensors together with microphones and video cameras is also forwarded to SocialFusion at regular intervals. The exact mechanism to facilitate this is dependent on the nature of the smart space.

Social networking data is obtained from online social networking websites such as Facebook, Twitter, LinkedIn, MySpace, etc.. In some cases, this requires permission from the users, e.g. Facebook. In others, the data is openly available, e.g. twitter, and can be imported using the standard APIs. Some information obtained from social networking websites, such as user preferences is relatively static, and is updated only once in a while. On the other hand, information such as frequency of communication between users, e.g. Wall posts, is obtained more often.

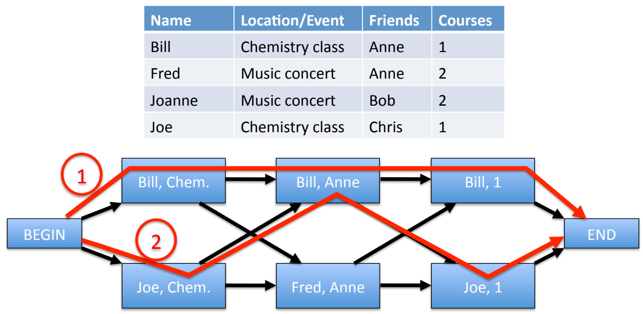
4.1 Protecting Privacy

A clear concern is protecting the privacy of information supplied by users of SocialFusion. We offer a multi-faceted solution. First, SocialFusion is predicated upon an opt-in approach. Users voluntarily opt into the system by downloading and installing the mobile application on their cell phones, agreeing during the sign up procedure to reveal their social networking and location information. Data uploaded from the mobile application is encrypted via https. Data accessed from social networks preserves the privacy policies of each of those networks. In the case of sites like Twitter, all information is public knowledge, while other sites like Facebook have elaborate privacy safeguards. However, even though SocialFusion adheres to the individual privacy policies, it is the combination of information across streams that both strengthens context and poses a greater challenge to privacy, i.e. correlating information about an individual across different sites and streams may collectively reveal more about an individual than that individual wanted to reveal when restricting access via independent filters on each site or stream.

To provide a more powerful safeguard for cross-stream privacy, we provide an anonymity layer that adheres to the classic K-anonymity criterion described earlier. The following challenges must be overcome in providing this anonymity guarantee: (1) accommodating the heterogeneity of the input mobile, sensor, and social data streams; (2) computing the anonymity criterion in near real time so that we can quickly decide whether the release of some data is K-anonymous - this is important so that the smart space can adapt quickly to the presence and preferences of a user; and (3) supporting subsequent stages of SocialFusion, namely the inference and recommendation stages.

Prior work on K-anonymity seems unsuited because it typically distorts the data before release either by introducing a random perturbation or transformation into the social graph [27, 33, 31], e.g. inserting a false node into a graph or a false link, or by generalizing or "fuzzifying" the information prior to public release [13], e.g. showing only an averaged value rather than a specific value. The problem this introduces is that distorting the information prior to inference can impair the quality of the inference. For example, inference that relies on identifying the leadership qualities of a group is very sensitive to the connectivity in the social graph, and introducing false information can mislead the inference and generate inaccurate descriptors. The ripple effect is that context-aware recommendations as well as output actions will also be negatively affected.

Instead, we have developed a new approach to K-anonymity based on the principle of *selective withholding*. In this approach, we do not distort the data in any matter, but rather decide only whether to selectively withhold the release of a piece of unperturbed data, determining if said data release compromises the K-anonymity guarantee. This approach is intuitively preferable to distortion, since describer modules and recommendation algorithms will operate only on clean original data, and in this way will not be misled with inferences based on falsified data, though in some cases



Name	Location/Event	Friends	Courses
Bill	Chemistry class	Anne	1
Fred	Music concert	Anne	2
Joanne	Music concert	Bob	2
Joe	Chemistry class	Chris	1

Figure 3: SocialFusion selectively withholds data in order to satisfy K-anonymity. A directed graph relating identity to the mobile, sensor, and social data generates a complex Boolean expression that is then minimized to find K.

they may only see a partial view of the total data picture in order to preserve K-anonymity. We believe that our selective withholding approach to K-anonymity breaks new research ground.

Our selective withholding approach to K-anonymity analyzes whether the release of a certain piece of data, e.g. a location or a social networking preference, will provide anonymity up to a degree of K. Consider the example shown in Figure 3. If the data set we wish to release is *(Chemistry class, Chris, 1)*, then the only possibility is *(Joe)*. However, if we withhold the data item *(Chris)* and only release *(Chemistry class, 1)*, then the possible identities are *(Joe)* OR *(Bill)*. Thus, we've increased the anonymity to $K \leq 2$. This example shows how we could selectively withhold certain data items to increase K to a desired value.

For more complex data sets, the Boolean expression linking the set of users to the set of data to be released needs to be systematically derived, which can be achieved using a directed graph that models the relationship between the data to be released and the users who could be linked to that data. For any data set that we wish to release (d_1, d_2, ..., d_n), we construct a column of nodes for each d_i, where each node in column i consists of the pair (username, d_i). This identifies all possible users who could be associated with the release of data item d_i. Next, we interconnect the nodes in a column with the nodes in the next column. This creates the directed graph. The set of all truth cases would be the superset of all paths from beginning nodes to the end nodes.

For example, before releasing *(Chemistry class, Anne, 1)*, we generate the directed graph shown in Figure 3 consisting of three columns, one each for all users associated with the Chemistry class, Anne, and one course. A single path #1 through the graph corresponds to the Boolean expression (Bill AND Bill AND Bill), or just (Bill). A second zigzag path #2 corresponds to the Boolean expression (Joe AND Bill AND Joe), or just (Joe, Bill). The union of all possible paths through the graph gives us all possible combinations of users that could be associated with the release of the data *(Chemistry class, Anne, 1)*.

Since the full Boolean expression could be quite complex, we need to devise an algorithm to quickly reduce the expression. We observe that logic minimization algorithms can accomplish this task. Several well known logic minimization algorithms exist, including ESPRESSO [11] and Quine-McCluskey. When applied to our example graph in the figure, the simplified Boolean expression elegantly reduces to (Bill) OR (Joe, Fred), thus illustrating the power of this technique. We determine if the simplified Boolean expression meets some K-anonymity guarantee by counting the number of terms in the simplified expression. The number of terms in the simplified expression equals the value K that we wish to compute.

5. PATTERN INFERENCE AND ACTUATION

Once mobile, sensor, and social data has been collected, stored efficiently and accessed through a K- anonymous interface the data can then be processed by the application. However, the application is confronted with a sea of raw data. SocialFusion uses "inference modules" to make sense of this "sea of data."

In general, the task of making sense of the sea of data consists of describing patterns found in the data. These patterns can then be used directly or help the application decide how to use the raw data. Pattern recognition must be done quickly and efficiently to support real-time applications. Also, the ability to find patterns that incorporate disparate data types such as correlating behavior information with location tags holds the most potential benefit for complicated mobile social networking applications (such as a real-time group based video recommendation system).

5.1 Frequent Pattern Analysis

Our first step is to identify frequent patterns in users' mobile, sensor, and social data, including both frequent itemsets and frequent sequences. A *frequent itemsets* is a set of items that co-occur frequently. For example, "John", "reading health-related news articles", "morning", and "riding bus to work" may be a set of items that co-occur many times in the mobile, sensor, and social data that we gather about John. A *frequent sequence* is a set of items that often occur in a certain sequence. For example, "pick up Tom", "go to a football game", and " having dinner" may be a sequence of activities (items) that appear frequently and in the same order for John and Mike. SocialFusion leverages existing frequent pattern analysis techniques developed in the data mining community. While traditional frequent pattern analysis techniques assume pre-defined and usually homogeneous items, SocialFusion has to support items from different data sources (e.g., mobile, sensor, social) and of diverse entities (e.g., user, content, time, location, activity), as well as easy and flexible extensions to support new types of items. In addition, different items have to be associated with their corresponding time information in SocialFusion, such that time-based ordering of items can be efficient calculated.

Due to the diversity in users' interests and daily activities, we first apply frequent pattern analysis techniques (both frequent itemsets and frequent sequences) on individual users and individual data sources of each user, then combine those patterns to infer more complex patterns that span across multiple users and multiple data sources. This approach helps us to quickly prune out large amounts of noisy and infrequent items in each category and focus on identifying and combining the much smaller number of patterns that are truly frequent and cover multiple users and data sources.

5.2 Pattern-Based Actuation

The frequent itemset and sequence patterns inferred above are indexed and maintained by the SocialFusion system, such that at run time, as up-to-date mobile, sensor, and social data are continuously captured, the SocialFusion system monitors the data and repeatedly conducts the following tasks:

1. Check the captured social fusion data against the inferred frequent patterns to determine if one or multiple matches exist between current state of data and frequent patterns.

2. If such a match does exist, the system determines further if an output action is needed and feasible, and the type of action (e.g., individual vs. group, playing a song vs. recommending a movie).

3. If the matched pattern indicates that a certain action is desirable, the system then invokes the corresponding actuation agent, such as choosing and playing a song that the user likes or recommending a movie to group of friends in the same room.

Checking for matched patterns and actuating the corresponding agents in SocialFusion are supported efficiently. Although large amounts of mobile, sensor, and social data continue to be captured and inserted into the SocialFusion system, since the system has already identified the small number of frequent patterns of interest and patterns that would require certain actuations, only certain contexts need to be detected and recommendations made, enabling very efficient output actions. Through the fusion of a variety of mobile, sensor, and social data, our SocialFusion system not only detects more reliable and more comprehensive patterns, but also enables more accurate context/activity classification and context-aware recommendations. Such capabilities and accuracies are not possible in previous systems that rely on single or same-type data sources. For instance, while a previous system may be able to recommend classic music and action movies to a user, SocialFusion could recommend classic music when a user is walking on the street or an action movie when he is with a group of friends on a Saturday evening.

5.3 Example: Real-Time Social Inference

A clear understanding or summary of social context is at the core of SocialFusion's Inference Engine. A set of inference modules have been implemented which can identify and describe social groups in real-time. Using trajectory information, possible "convoys" are identified and then a particular query (group or user) is matched to a "convoy" optimizing for social and accelerometer indicators such as social network friendship, correlated movements and physical orientation. This convoy is then efficiently described as it relates to the particular query. For instance, if the query targets movie information, an "expert descriptor" would be included with the group description implying the relative expertise of group members as related to movies.

Group identification.

This can be framed as a frequent itemset problem. For instance, when location updates are searched within an appropriate temporal window trajectory clustering or "convoy detection" effectively becomes a multi- dimensional clustering problem in which time-stamped locations are clustered within different segments of time and frequently occurring clusters represent convoys. This can be done through maintaining and modifying a "fuzzy" set of clusters across multiple time periods as done for object tracking in many applications [24]. Because SocialFusion uses many different data sources including phone sensors and social information to more precisely identify social structure, fuzzy clustering works particularly well for SocialFusion's group identification as it is straightforward to perturb the clusters. For example, accelerometer readings from multiple phones can be correlated to infer grouping [34]. Phone sensors can not only be correlated with each other but can also be associated with simple group behaviors or actions which give us more information about the state of the group. Furthermore, groupings can be optimized by evaluating the social relations between group members such as the friend-link density between users of Facebook or the temporal contact graph gathered over long periods of time which could detect long-term patterns such as meetings that occur every third monday of the month.

The general approach taken by this project is to first do fast and efficient trajectory clustering or convoy detection resulting in a fuzzy clustering of all nearby clusters relevant to a query or individual. The relevant indi-

Table 1: Classification of Group Descriptors

	content	social structure
single user	Expert	Leader
pair of users	Similarity	Social

vidual or individuals are then checked for other available sources of information (Facebook, sensor data, long-term behavior modeling) with which the clusters are modified and the maximally relevant cluster is chosen.

Group descriptors.

In SocialFusion, a set of group describer modules are implemented to capture the key characteristics of a group and use these characteristics for group-based actions (e.g., movie recommendation). Usually, multiple group describers can be used, and each group describer aims to characterize or summarize a particular aspect of a group. Specifically, the following group describer modules have been implemented in SocialFusion:

Expert Describer measures the relative expertise of different group members as related to a specific topic. The opinions of experts may be weighted more important than that of other group members. The describer also summarizes a group's absolute mean expertise, e.g., a group of casual movie watchers or film critics.

Leader Describer measures the influence of individual members in a group based their social networks using degree or betweenness centrality metrics. For example, a person who has more friends in his/her social networks and posts more frequently is considered a leader and more influential in a group. The describer also summarizes how unbalanced the group members, such as a group of equally-important peers or a hierarchy centered around a few group members.

Social Describer measures the strength of all pairwise member social links in a group. Tighter and stronger social relationships can be more influential within a group. Also, the describer summaries how densely or strongly a social group is connected, such as a group of highly-connected close friends or relative strangers.

Similarity Describer provides a relative weighting of how similar each pair of users are in terms of their content interests. Closely-shared interests can be more influential in a group. Also, the general interest diversity of an entire group can be summarized by this describer.

As shown in Table 5.3, the group descriptors generated by the four describers above characterize groups along two major dimensions: (a) pairwise relations vs. individuals; and (b) content vs. social structure. Both Leader and Expert are single-user descriptors, while Similarity and Social are pairwise descriptors. On the other hand, both Expert and Similarity are content-based descriptors, while Leader and Social are based on social structures. Depending on a group's activity state or purpose and how cohesive the group is, certain descriptors may weight more than the others. For instance, in some situations, experts may be more important and in other situations social relations may dominate group decisions. Our current work in SocialFusion is to develop and integrate useful metrics that not only describe a group but also guide how descriptions of the group should be interpreted and utilized for upper-level applications.

6. CASE STUDY: GROUP-BASED MOVIE RECOMMENDATION SYSTEM

To validate our design of the SocialFusion framework and to demonstrate how SocialFusion enables more powerful context-aware mobile computing, we present in this section a case study of a group-based movie recommendation system. This case study makes specific use of the "fusion" capabilities of SocialFusion, capturing important mobile, sensor, and social data, inferring inherent group-based patterns and characteristics, and using the inferred patterns for group-based actuation, i.e., recommending movies to a group of users who are physically located in the same room.

Although various recommendation systems exist for individual users [6], groups of users [20], and specific data types, SocialFusion is the first framework that enables group recommendation based on comprehensive mobile, sensor, and social data. Instead of a simple approach that aggregates multiple users into a single "virtual" user, SocialFusion extracts and makes use of a number of group-based descriptors for better informed group recommendation. We have described in the previous section how a coherent social group can be identified and how different group descriptors can be extracted from a group. In this section, we describe in detail the content-based group recommendation technique and collaborative-filtering-based group recommendation technique for group-based movie recommendation, focusing on how different group descriptors may be utilized in the recommendation process to achieve the best group overall satisfaction.

6.1 Content-Based Group Recommendation

Content-based recommendation techniques are based on the assumption that if a user likes item A, then she will like other items that are similar to item A. In the case of movies, by analyzing the attributes of different movies (e.g., director, actors) and their similarities, movies of similar attributes can then be recommended based on users' movie viewing history. To make content-based group recommendation using SocialFusion, we need to overcome the following key challenges: (1) obtaining individual members' movie preferences; (2) generating group-based movie preferences; and (3) identifying dominating movie attributes for similarity measure. We address these challenges as follows.

First, to obtain the movie preferences of individual members in a group, we examine these users' Facbook profiles, which are automatically captured and integrated into the SocialFusion framework. A user's Facebook profile usually contains a favorite movie list which describes the movies that the user likes the best. However, these are just *positive* examples. To fully capture a user's movie interest, we also need to obtain *negative* examples, i.e., the movies that a user does not like. To address this problem, we introduce a mapping scheme that makes use of the Netflix data set [4], which contains over 100M ratings of nearly 18K movies by over 480K users. On average, each user rates over 200 movies using the scale of 1 to 5 (5 is best). Due to the difference in user's rating criteria, we normalize each user's ratings as follows:

$$R' = \frac{R - U_{min}}{U_{max} - U_{min}}. \quad (1)$$

where R' and R are the normalized and original ratings; U_{max} and U_{min} are the maximum and minimum ratings of that user. Using the favorite movies listed in a Facebook user's profile, we then map that user to a Netflix user by computing the Jaccard similarity coefficient between the Facebook user and a Netflix user and pick the most similar Netflix user. In a real usage scenario, it is possible for SocialFusion to automatically integrate a user's Facebook account and Netflix account.

Next, we combine the movie preferences of individual group members to generate a movie preference for the entire group. We select the movies which have been rated by most of the group members. More importantly, we harness two group-based satisfaction metrics: *maximizing satisfaction* and *minimizing misery*. Let G be a user group, and $r_{u,m}$ be the rating of user u on movie m. Then the group-based satisfaction metrics are

$$r_{max-satisfaction}(G, m) = \frac{1}{|G|} \sum_{u \in G} r_{u,m} \quad (2)$$

$$r_{min-misery}(G, m) = \min\{r_{u,m} | u \in G\} \quad (3)$$

Based on the group ratings, movies are divided into two categories: the ones that the group like or dislike. The threshold is set to be the mean plus standard deviation of all the movie ratings by the group.

Given a group's liked and disliked movies, we obtain the "plot key words" for each movie from a local copy of the IMDB data set [1]. Since not all plot key words are important in terms of classifying the liked and disliked movies, we compute the expected information gain $IG(w, M)$, which is the "classification power" of plot key word w in a set of movies M, i.e., whether the presence or absence of w helps us to determine if the group likes a movie or not:

$$IG(w, M) = I(M) - \left(\frac{|M_w|}{|M|} \cdot I(M_w) + \frac{|M_{\bar{w}}|}{|M|} \cdot I(M_{\bar{w}}\right) \quad (4)$$

[1]IMDB `http://www.imdb.com/interfaces`

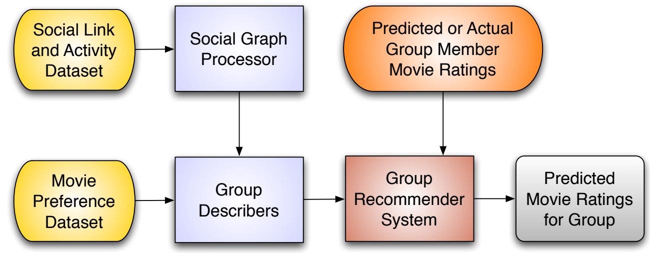


Figure 4: Collaborative-filtering-based group recommendation workflow

where $I(M)$, $I(M_w)$, and $I(M_{\bar{w}})$ are the entropy of the set of all movies, movies containing w, and movies not containing w, respectively. Here are some of the most informative plot key words obtained from a collection of 80 movies:

Airplane Accident, Child, Tragic Villain, WWII, 1930s, 1940s, Monkey, Single Mother, Mental Illness, Nightclub, Compassion, Crushed To Death, Disney Animation Feature, Technology, ...

Once we have identified the most informative plot key words, we can then make group movie recommendation using standard machine learning techniques, including ID3, Naive Bayes and Support Vector Machine (SVM).

6.2 Collaborative-Filtering-Based Group Recommendation

SocialFusion also uses collaborative filtering heuristics to generate recommendations for groups of users. We first use a standard collaborative filtering system, like Apache Mahout Taste, to generate individual predicted movie ratings for each group member for a specified set of movies. After obtaining these individual movie ratings, we compute predicted movie ratings for the group using several different heuristic functions. We have implemented five different recommendation methods. Figure 4 depicts the general workflow. In each method, we use an aggregation function to compute a predicted movie rating for the group of users based on the individual predicted or actual movie ratings of each group member. The first four methods use each of the four descriptors (leader, expert, social, and similarity) independently. The fifth method, called the all-descriptors recommender, determines a predicted rating for a group of users by computing the weighted sum of the predicted group ratings output from the first four recommender systems.

A weighted sum function is used to compute a predicted group rating in the leader and expert based recommendation methods:

$$r_{G,m} = k \sum_{u \in G} desc(u) \cdot r_{u,m} \quad (5)$$

where $r_{G,m}$ is the value of the predicted rating for group G and movie m, user u is a member of G, $desc(u)$ is the descriptor value for u, and $r_{u,m}$ is u's predicted rating for v. The multiplier k serves as a normalizing factor and is defined as $k = 1/\sum_{u\in G} desc(u)$. Note that this function is similar to the aggregation functions used in many heuristic-based (memory-based) collaborative filtering systems [6].

Describers that measure the relative strength of relations between users need a way to emphasize the ratings of two users in some way that would affect the expected group rating. If the relationship is very strong, the shared rating of the users would be pushed away from the mean (toward the extremes of the rating range) and if the relationship is weak the shared rating would be pushed toward the mean. This is accomplished by using a specialized sigmoid function which closely resembles a Gompertz curve centered around the mean:

$$f(x) = \frac{S}{1+e^{-wt}} \tag{6}$$

where S is the magnitude of the rating scale, w is an appropriate weight shaping the sigmoid function and

$$t = \left(\frac{r_1 r_2}{r_{mean}^2}\right)^{\frac{s_{12}}{s_{mean}}} \tag{7}$$

The variable t allowed the integration of user ratings (r_1, r_2) and the users' relationship strength s_{12} in such a way that when user ratings are greater than the mean rating r_{mean} then the base of the exponent is greater than 1.0 and conversely when the ratings are less than r_{mean} the base is less than 1.0. Since this base is raised to the users' relationship strength s_{12} over the mean relationship strength s_{mean} then the overall value of t is further from 1.0 when the users' relationship strength is greater than average and is nearer to 1.0 when the users' relationship strength is less than average. Therefore the mean of many values of the sigmoid function will be affected more by the fused ratings of users with a strong relationship than the fused ratings of users' with weak relationships.

The similarity and social recommender systems both use the sigmoid function described above to compute a predicted movie rating for a group:

$$r_{G,v} = \frac{\sum_{u_1,u_2\in G} sigmoid(desc(u_1,u_2), r_1, r_2)}{|Pairs(G)|} \tag{8}$$

where $sigmoid(desc(u_1,u_2), r_1, r_2)$, $desc(u_1,u_2)$ is the value of the similarity or social descriptor between users u_1 and u_2, and $|Pairs(G)|$ is the number of pairs of users in G with similarity/social descriptor values.

The all-descriptors recommender determines a predicted movie rating for a group by computing the weighted sum of the predicted movie ratings generated from the four single-descriptor recommender systems:

$$\begin{aligned} r_{G,v} = a \cdot r_{G,v,expert} + b \cdot r_{G,v,leader} + \\ c \cdot r_{G,v,similarity} + d \cdot r_{G,v,social} \end{aligned} \tag{9}$$

where $r_{G,v,descriptorName}$ is the group movie rating predicted by each single-descriptor recommender system. We determine the optimal weights a, b, c, d for a group by performing a brute-force iterative search to find the weights which produce the minimum RMSE value.

7. EVALUATION

In this section, we conduct a set of evaluations to validate the design of the SocialFusion system and demonstrate its effectiveness in terms of capturing diverse group characteristics and facilitating context-aware and group-based recommendation. Specifically, we start by describing the implementation of the mobile system and an evaluation of the K-anonymity algorithm. We then present a detailed analysis of various group descriptors using the Facebook and Netflix data sets. We continue with a real user group study and some key observations. Finally, we present the evaluation results of our content-based and collaborative filtering based group recommendation techniques.

7.1 Mobile Client Implementation

We have designed and implemented a mobile client that allows users to anonymously share their presence [9] via Bluetooth. This mobile system is composed of a mobile component (MC) that resides on a mobile device and a stationary component (SC) that resides on a desktop or laptop PC. The MC, implemented in Java ME, allows a user to log in to his/her Facebook account and initiate anonymous sharing of his/her Facebook ID. The SC, implemented in Java SE, detects a user's shared Facebook ID and uses it to submit queries to SocialFusion, such as retrieving the user's movie preferences or accessing the user's social networks or Facebook wallposts. The MC has been tested on Nokia N95 smartphones. We are currently working on a new version of the MC that will run on Nokia N97 smartphones and share sensor information from the accelerometer, GPS, magnetometer (compass), and camera with SocialFusion.

7.2 K-Anonymity

We have implemented a prototype of SocialFusion's selective withholding K-anonymity algorithm and performed an initial evaluation of its behavior and scalability. Selective withholding consists of two components, namely the Boolean expression builder, and then the logic minimizer component that yields the value K. We use an open source Quine-McCluskey implementation [5] to perform logic minimization over the Boolean expression generated from the directed graph. We ran the tests on a Macbook using a university Internet connection.

Our metrics were gathered using the Facebook account of one of the participants, here called "user A", who has 222 Facebook friends and seven favorite movies listed on his Facebook profile. A study has shown that the mean number of Facebook friends reported by the study participants was between 150 and 200 [17] . Therefore, we suppose that user A is a reasonable representation of a typical or average Facebook user.

We conducted our evaluation of K-anonymity performance by having SocialFusion submit a query to Facebook requesting the list of favorite movies in the Facebook profile for user A. SocialFusion also gathers the favorite movies lists for each of user A's Facebook friends. SocialFusion then proceeds to construct a Boolean expression representing the relationship between user A's friends and their favorite movies.

Figure 5 shows how the time required to minimize the unsimplified Boolean expression in the SocialFusion logic minimizer component varies with the number of terms in the unsimplified Boolean expression. We see from this plot that the logic minimizer component scales reasonably well for unsimplified Boolean expressions containing up to about 450 terms. The maximum number of terms in the unsimplified Boolean expression that we saw from this data set of Facebook users and favorite movies was 450 terms. We expect that 450-term Boolean expressions will account for many typical usage scenarios, although this will vary based on the number of the user's favorite movies that match with friends' favorite movies. There is a nonlinear relationship between the number of terms in the unsimplified Boolean expression and the number of terms in the simplified expression. Based on the results of our tests, we have found that unsimplified Boolean expressions containing around 450 terms have up to about 20 terms when simplified by the SocialFusion's logic minimizer. 20 terms in the simplified Boolean expression provides K-anonymity guarantees for $k = 20$. Thus, we have shown that SocialFusion's selective withholding is feasible for K-anonymity guarantees up to $k = 20$, which includes user groups as large as most social network friend lists (consisting of 200–300 friends).

Table 2 shows the run times of each component of SocialFusion for the following conditions for user A: number of friends = 222, number of movie matches = 13, number of friend matches = 7, number of terms in unsimplified expression = 339, number of terms in simplified expression = 7. The SocialFusion total run time in table 2 is the time for our system to return K-anonymous favorite movie preferences for a user. In our tests, the run time of the social network data gatherer component dominates the run time of SocialFusion. We expect that running SocialFusion with local access to a user database would significantly reduce the run time of this component. However, the current average total SocialFusion run time of 1377 ms for our tests provides acceptable performance for applications such as context-aware video playback.

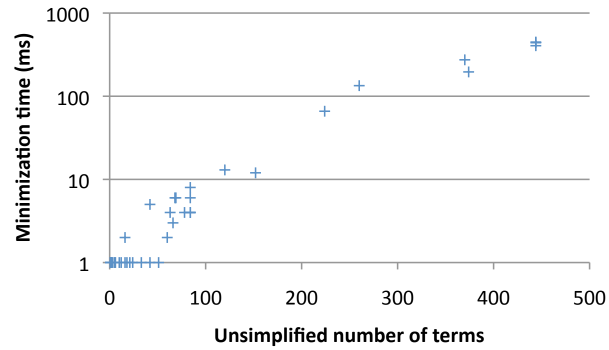


Figure 5: Time to minimize the unsimplified Boolean expression vs. number of terms in the unsimplified Boolean expression

Table 2: Run times of SocialFusion Components

Component	Mean	Std-Dev (σ)
data gatherer	852 msec	145 msec
expression builder	11 msec	1.7 msec
logic minimizer	297 msec	8.29 msec
total	1377 msec	134.7 msec

7.3 Group Descriptors

To evaluate the different types of group descriptors, we analyze a publicly available Facebook data set composed of over 60,000 Facebook users [32]. We selected several groups by randomly choosing a starting user node in the social graph constructed from this data set and then performing a breadth-first traversal to find connected nodes. Figures 6 and Figure 7 show social graphs for two groups composed of five and ten users, respectively.

Summary group descriptor values of the two groups are shown in Figure 8. The Expert summary value is computed as $\sum_{u \in G} |M_u|/|G|$, where $|M_u|$ is the number of favorite movies user u has. The Leader summary value is computed as $\frac{max(|N_u|)}{(\sum_{u \in G} |N_u|)/|G|}$, where $|N_u|$ is the number of friends u has in group G. The Similarity and Social summary values are computed as $\frac{\sum_{u_1,u_2 \in G} desc(u_1,u_2)}{|Pairs(G)|}$, where $desc(u_1, u_2)$ is the value of the Similarity or Social descriptor between users u_1 and u_2 in G, and $|Pairs(G)|$ is the number of user pairs in G with Similarity or Social descriptor values. From Figure 8, we can make the following observations between the 5-user group and the 10-user group: (1) the 5-user group has more expertise; (2) the 10-user group is more leader-based; (3) the 5-user group is more socially cohesive; and (4) the 10-user group has some users with similar movie interests. These results demonstrate the heterogeneity of groups and the importance of capturing group characteristics for better context-aware and group-based actions.

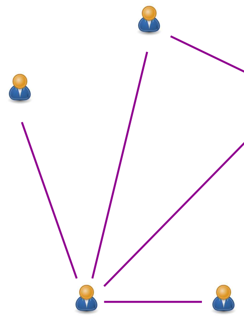


Figure 6: A social group composed of five Facebook users.

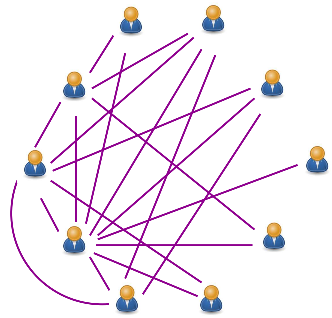


Figure 7: A social group composed of ten Facebook users.

Group size	Group descriptor	Summary value
5-user	Expert	8.20
5-user	Leader	2.00
5-user	Similarity	0
5-user	Social	1.20
10-user	Expert	1.90
10-user	Leader	2.37
10-user	Similarity	0.0177
10-user	Social	1.0

Figure 8: Group descriptor summary values of social groups.

7.4 Group Study of Real Users

To understand how a group of people reaches a decision and how the group decision may be affected by various group characteristics, we have conducted a real user group study consisting of 12 participants. There are 9 males and 3 females in the group, aged between 18 and 30, and with different ethnic backgrounds. Our study was conducted in two stages.

In the first stage, the participants were asked to rate 100 movies on a scale of 1–5 or indicate that they had not seen a movie. On average, each participant has seen and rated 68.6% of the 100 movies, and each movie has been rated by at least 70.0% of the participants. Each participant was also asked to specify how often he/she interacts with other participants in the group. Each participant knows some members in the group but not all of them, thus forming a connected social graph with multiple hops between certain members. In the second stage, all participants gathered in the same room and rated 20 movies. Each movie is rated as follows: (1) show the movie trailer; (2) participants write down their own rating of the movie; (3) participants discuss the movie and vote on a single group rating for that movie; and (4) participants write down how they feel about the group decision and why they feel that way.

We make several important observations based on our real-user group study. First, certain individuals interacted much more than other members of the group, and their opinions seemed to influence other members in the group – a pattern captured by our *Leader Describer*. Second, certain members had not seen the movies and seek the opinion of others who have seen the movie or many other movies – such expert-based effect is captured by our *Expert Describer*. Third, when asked how they feel about the group decisions, some participants pointed out that other members had similar tastes to their own and that their opinion was highly affected by these similar participants – this is captured by our *Similarity Describer*. It was also observed that certain members of the group knew each other very well and tended to talk directly to each other, forming a shared opinion in the group – such social relationship based effect is captured by our *Social Describer*.

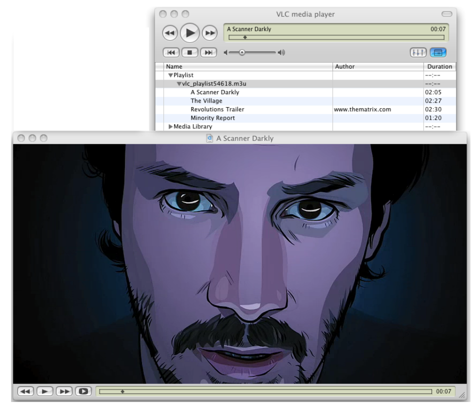


Figure 9: Screenshot of a SocialFusion's video application playing a recommended film.

7.5 Group-Based Movie Recommendation

Next, we evaluate the performance of group-based movie recommendation, using either content-based or collaborative filtering based group recommendation methods. To evaluate the content-based methods, we use the individuals' movie ratings obtained in the first stage of the real user group study as the training data, and three supervised learning algorithms (ID3, NaiveBayes, and SVM). We consider two group-based prediction methods: maximizing-satisfaction (i.e., maximizing mean value of predicted ratings) and minimizing-misery (i.e., maximizing the minimum predicted rating). To evaluate the collaborative filtering based methods, we use individual users' movie ratings to predict their group ratings, utilizing one or all the four group descriptors: leader, expert, similarity, and social relationship. Figure 9 shows a movie trailer recommended to a SocialFusion user.

Content-based group recommendation.

Table 3 shows the precision and recall of different content-based methods. SVM achieves better perfor-

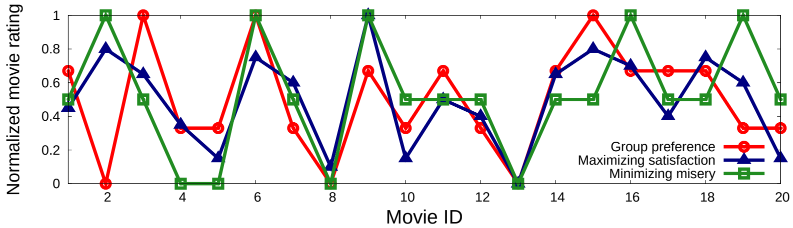


Figure 10: Comparison of actual group ratings, maximizing-satisfaction, and minimizing-misery.

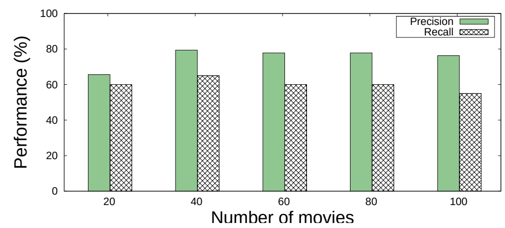


Figure 11: Performance vs. number of movies in the training set.

Table 3: Performance of content-based group recommendation methods

		ID3	NaiveBayes	SVM
maximizing satisfaction	precision	50.0%	25.0%	65.6%
	recall	30.0%	50.0%	60.0%
minimizing misery	precision	50.0%	50.0%	76.3%
	recall	50.0%	10.0%	55.0%

mance than ID3 and NaiveBayes. More importantly, minimizing misery achieves better group recommendation than maximizing satisfaction. This is also demonstrated in Figure 10. The Pearson correlation coefficient between maximizing-satisfaction and actual group rating is only 0.5927, while minimizing-misery achieves 0.9485 correlation with actual group rating. This can be explained by the relatively sparse group structure (social descriptor value of 118.6 vs. maximum score of 365.0), i.e., the members in this group are less cohesive than average (half of the max or 180.0). In this case, the users are more like strangers and minimizing-misery tends to be the dominating effect. We further investigate whether fewer number of movies in the training set degrades the recommendation performance. As shown in Figure 11, when the number of training movies decreases from 100 to 20, there is no significant performance degradation in terms of precision and recall.

Collaborative filtering based group recommendation.

In the real user group study, we obtained both movie ratings of individual users and group movie ratings. To evaluate the collaborative filtering based group recommendation methods we have developed, we take the movie ratings of individual users and predict the group rating using different group descriptors, then compare the predicted group rating to the actual group rating. The metric we use is *root mean squared error* (RMSE):

$$RMSE = \sqrt{\frac{\sum_{m \in M}(r_{G,m} - r'_{G,m})^2}{|M|}} \tag{10}$$

which measures the difference between group G's actual rating $r_{G,m}$ and predicted rating $r'_{G,m}$ for each movie m in a set of movies M. Table 4 shows the RMSE values of the baseline (mean-based) approach and all five of our collaborative filtering methods. Of the single-descriptor approaches, the similarity recommender has the best

Table 4: Performance of collaborative filtering based group recommendation methods

Recommender	**RMSE**	**Improvement over baseline**
mean-based (baseline)	0.7658	–
social	0.9304	-21.50%
leader	0.8267	-7.95%
expert	0.7554	1.36%
similarity	0.7199	5.99%
all-descriptors	0.7186	6.16%

accuracy. The all-descriptors approach achieves even better accuracy, with the optimal weights $a = 0.0168$, $b = 0$, $c = 0.92$, and $d = 0.0632$. These weights show that for this group, user similarity (c) has the most impact on the group's movie ratings, while social relationship (d) has a small but noticeable impact. This confirms our observation that content similarity was dominant in our user study group, which consisted of loosely-connected individuals.

The 6.16% RMSE improvement over the baseline would be significant in the Netflix Prize context [2], though our problem setting differs in its scale and group focus.

8. FUTURE WORK

We plan to integrate more components into SocialFusion, including input from fixed sensors, a hybrid recommendation engine, and the anonymization layer with its privacy and security components. We also plan to explore other context-aware multimedia applications beyond our case study. We intend to conduct further user studies to evaluate SocialFusion's usability, the impact of its recommendations on users' decisions, and how privacy affects users' attitudes towards SocialFusion. Scalability and performance are other areas that need more investigation. We further plan to incorporate an API so that other context-aware applications can easily build upon our framework. We also will release SocialFusion as open source for the research community, so that other researchers can add their own modules for inference, recommendation, and context-aware experimentation.

[2] `http://www.netflixprize.com/community/viewtopic.php?id=828`.

9. CONCLUSIONS

This paper has presented SocialFusion, a new framework for fusing together mobile, social and sensor networks in order to support the next generation of increasingly sophisticated context-aware applications. SocialFusion consists of 3 stages: first, a data gathering and management stage, including a novel K-anonymization algorithm; next, an inference stage that fuses together the diverse data streams using describer modules to extract contextual clues called descriptors; finally, a recommendation stage that leverages the rich assembled data and descriptors to recommend a context-aware action. A case study of a group-based context-aware video application illustrated how SocialFusion can improve context awareness, especially for groups of people.

10. REFERENCES

[1] Brightkite. `http://brightkite.com`.

[2] Loopt. `http://www.loopt.com`.

[3] Mobile apps help people track h1n1, prevent flu, nov 25, 2009, usatoday. `http://www.usatoday.com/news/health/2009-11-25-new_fluapps23_ST_N.htm`.

[4] Netflix. `http://www.netflix.com`.

[5] Quine-mccluskey algorithm (java). `http://en.literateprograms.org/Quine-McCluskey_algorithm_(Java)`.

[6] 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.

[7] M. Balabanovic and Y. Shoham. Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3), 1997.

[8] A. Beach, M. Gartrell, S. Akkala, J. Elston, J. Kelley, K. Nishimoto, B. Ray, S. Razgulin, K. Sundaresan, B. Surendar, M. Terada, and R. Han. Whozthat? evolving an ecosystem for context-aware mobile social networks. *IEEE Network*, 22(4):50–55, July-August 2008.

[9] A. Beach, M. Gartrell, and R. Han. Solutions to security and privacy issues in mobile social networking. In *Workshop on Social Mobile Web (SMW09), associated with SocialCom09*, 2009.

[10] D. Billsus, Pazzani, and J. Chen. A learning agent for wireless news access. In *The International Conference on Intelligent User Interfaces*, 2002.

[11] R. K. Brayton, A. L. Sangiovanni-Vincentelli, C. T. McMullen, and G. D. Hachtel. *Logic Minimization Algorithms for VLSI Synthesis*. Kluwer Academic Publishers, Norwell, MA, USA, 1984.

[12] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4), 2002.

[13] A. Campan and T. Truta. A clustering approach for data and structural anonymity in social networks. In *PinKDD 2008*. ACM, August 2008.

[14] L. P. Cox, A. Dalton, and V. Marupadi. Smokescreen: flexible privacy controls for presence-sharing. In *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 233–245, New York, NY, USA, 2007. ACM.

[15] N. Eagle and A. Pentland. Social serendipity: Mobilizing social software. *IEEE Pervasive Computing*, 4(2), April-June 2005.

[16] W. K. Edwards and R. Grinter. At home with ubiquitous computing: Seven challenges. In *Proceedings of the 3rd International Conference on Ubiquitous Computing (Ubicomp 2001)*, pages 256–272, May 2001.

[17] N. Ellison, C. Steinfield, and C. Lampe. The benefits of facebook "friends:" social capital and college students' use of online social network sites. *Journal of Computer-Mediated Communication*, 12(4), 2007. `http://jcmc.indiana.edu/vol12/issue4/ellison.html`.

[18] I. Garcia, L. Sebastia, E. Onaindia, and C. Guzman. A recommendation system for tourist activities. *LNCS: E. Commerce and Web Technologies*, 5692, 2009.

[19] C. M. Gartrell. Socialaware: Context-aware multimedia presentation via mobile social networks. Master's thesis, University of Colorado at Boulder, December 2008.

[20] A. Jameson. More than the sum of its members: challenges for group recommender systems. In *Proc. of the working conf. on Advanced visual interfaces (AVI 2004)*, pages 48–54, Gallipoli, Italy, May 2004.

[21] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen. Discovery of convoys in trajectory databases. *Proc. VLDB Endow.*, 1(1):1068–1080, 2008.

[22] T. Kindberg and J. Barton. A web-based nomadic computing system. Internet, August 2000.

[23] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3), 1997.

[24] P. X. Liu, M. Meng, and C. Hu. On-line data-driven fuzzy clustering with applications to real-time robotic tracking. In *2004 IEEE International Conference on Robotics and Automation*, April 2004.

[25] S. Loeb. Architecting personal delivery of multimedia information. *Communications of the ACM*, 35(12), 1992.

[26] J. Manweiler, R. Scudellari, Z. Cancio, and L. P. Cox. We saw each other on the subway: secure, anonymous proximity-based missed connections. In *HotMobile '09: Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, pages 1–6, New York, NY, USA, 2009. ACM.

[27] G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks. Technical Report Technical Report 07-19, Computer Science Department, University of Massachusetts Amherst, March 2007.

[28] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell. Cenceme - injecting sensing presence into social networking applications. In *Proc.of the 2nd European Conf. on Smart Sensing and Context (EuroSSC 2007)*, Oct. 2007.

[29] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proc. of the 6th ACM Conf. on Embedded Network Sensor Systems (SenSys 2008)*. ACM, Nov. 2008.

[30] M. O'Connor, D. Cosley, J. Konstan, and J. Riedl. Polylens:a recommender system for groups of users. In *The European Conference on Computer-Supported Cooperative Work*, 2001.

[31] B. Thompson and D. Yao. The union-split algorithm and cluster-based anonymization of social networks. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (ASIACCS 2009)*, pages 218–227. ACM, March 2009.

[32] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09)*, August 2009.

[33] Q. Wei and Y. Lu. Preservation of privacy in publishing social network data. In *Proceedings of the 2008 International Symposium on Electronic Commerce and Security (ISECS 2008)*, pages 421–425. IEEE Computer Society, March 2008.

[34] M. Wirz, D. Roggen, and G. Tröster. Decentralized detection of group formations from wearable acceleration sensors. In *Proceedings of the 2009 IEEE International Conference on Social Computing*, page IEEE Press, 2009.