Knowledge-Based Systems 131 (2017) 70–82

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A smartphone-based activity-aware system for music streaming recommendation

Wei-Po Lee*, Chun-Ting Chen, Jhih-Yuan Huang, Jhen-Yi Liang

Department of Information Management, National Sun Yat-sen University, Kaohsiung, Taiwan

ARTICLE INFO

Article history:
Received 9 August 2016
Revised 8 April 2017
Accepted 2 June 2017
Available online 3 June 2017

Keywords:
Activity recognition
Context-awareness
Mobile music recommendation
Feature extraction
Classification
Smartphone

ABSTRACT

Contextual information is helpful in building systems that can meet users' needs more efficiently and practically. Human activity provides a special kind of contextual information that can be combined with the perceived environmental data to determine appropriate service actions. In this study, we develop a smartphone-based mobile system that includes two core modules for recognizing human activities and then making music streaming recommendation accordingly. Machine learning methods with feature selection techniques are used to perform activity recognition from smartphone signals, and collaborative filtering methods are adopted for music recommendation. A series of experiments are conducted to evaluate the performance of our activity-aware framework. Moreover, we implement a mobile music streaming recommendation system on a smartphone-cloud platform to demonstrate that the proposed approach is practical and applicable to real-world applications.

1. Introduction

In recent years, context awareness has played an important role in deploying high-quality information services [3,6]. It involves capturing a broad range of contextual attributes to determine services on demand. By integrating context information into an application service, a system can fulfill a user's needs more efficiently and practically through continuously adapting to the dynamic situational changes in the user's environment and in his intrinsically affective and cognitive states. Considering the advantages of contextual factors, in this work we develop an activity-aware system for music streaming recommendation.

Human activity provides a special kind of contextual information. It can be combined with the perceived environmental data to form a complete world state representation and can be used to determine appropriate service actions. Several steps are involved in human activity recognition (HAR): recording a user's behavior for a period of time, extracting the relevant information from the behavior sequences, and analyzing the sequence data to derive specific patterns. These patterns reveal how users prefer to consume the specific services, and the patterns can be used to make recommendations more suitable for them. To perform activity recognition for mobile users, different wearable sensor techniques have been proposed to work with computational methods. Smartphones are currently the most representative portable products in the evolution of information and communication technologies [15,31]. These self-contained handheld devices have advanced features, such as mobile operating systems, broadband internet access and other computer-like processing capabilities, and these features make the devices highly suitable for activity recognition. Therefore, in this study we explore the use of smartphones as an alternative approach for identifying physical activities. The information obtained can be used as input for a music streaming recommendation service. Time series data collected from different ubiquitous sensors can be used to recognize user activities. Nevertheless, the limited sensing ability and the intrinsic noisy character of the sensory data restrict the types of activities identifiable.

Regarding music recommendation, it is particularly important to take into account contextual aspects of the user in a mobile music consumption scenario, in which the user context frequently changes. Some psychological and sociological studies on music and people show that the users' short-term needs (in contrast to long-term needs, such as user preferences) are usually influenced by their context, such as their emotional states, activities, or external environments [27,48]. For instance, a user usually prefers loud energy-boosting music when running and mood music when trying to sleep. Recently, considerable attention has been directed towards context-aware music recommender systems to utilize contextual information for making better recommendations. However, although existing systems explore many kinds of explicit context information (such as location, time, weather, and running pace [20,33]), they hardly consider daily activities (such as walking, sleeping, running, and studying). It is well acknowledged that peo-

* Corresponding author.
E-mail address: wplee@mail.msysu.edu.tw (W.-P. Lee).

http://dx.doi.org/10.1016/j.knosys.2017.06.002
0950-7051/

ple prefer different kinds of music for different daily activities. An adaptive music system is able to detect users' daily activities in real-time and play suitable music automatically; it can thus satisfy the users' needs as well as save time and effort in choosing suitable music.

In this study, we develop a new pervasive smartphone-based system to address the two above-mentioned issues (i.e., activity recognition and music recommendation) to make appropriate music streaming recommendations. Here, the recommendation corresponds to the preparation of a list of music from a music streaming site. Our work adopts machine learning algorithms to recognize a user's activities and to estimate the user's music preference in various situations. Due to the differences in the contextual information and the datasets used, the performance of different works cannot be compared in a direct way. Therefore, comparisons are made of different methods for extracting the features of time and frequency domains from the smartphone signals, and the effects on different types of daily activities are examined. Moreover, a mobile client and cloud server architecture is presented to reduce computational load and enhance data management. To verify the proposed activity-aware framework, we conduct a series of experiments for performance evaluation. The results show that the accuracy of different methods varies from 66.4% to 83.4% in activity recognition, and from 71.5% to 77.6% in music recommendation. In addition, to prove that the proposed approach is more practical and applicable to real-world applications, we implement a prototype system for activity-aware music recommendation.

2. Background and related work

In the development of context-aware service systems, various contextual information has been proposed and applied for decision-making; see, for Example [38,50]. The information often used is based on smartphone data, where the extracted information is utilized to develop location-aware services with real-time estimated user locations. For Example, Kim et al. proposed the development of smartphone-based systems to provide location information that could be combined with GPS and Wi-Fi positioning systems in order to generate user contexts. These contexts could then be used to build location-based services in daily life [32]. In addition, there has been growing interest in developing proactive wellness products and health-related smartphone applications, such as user-adapted fitness games or physical fitness activities [1,28]. As shown in [10,13,45], a mixture of behavioral, spatial and social information obtained from smartphones has been used to construct cyber-physical world applications, in which location-based social networks are developed and applied to make location or activity recommendations. In this work, we also use smartphones to collect and tailor sensory information to represent users' context.

When building up an activity-aware music streaming system, we have to consider two important issues: how a context-aware music system is built and how the user activities are recognized. In the following, we analyze relevant studies from the above two perspectives. Context-aware music recommender systems (CAMRSs) utilize contextual information and provide better user-centered services such as song recommendation. Regarding context-aware music recommendation, related works mainly differ in how user context information is defined, gathered and used. In general, the contextual information can be classified into two types: environment-related and user-related context information [21,29]. The environmental context describes the set of a user's external circumstances, such as the user's location and the current time, weather, and temperature. In applications with this type of context for music recommendation, researchers explored the possibilities of adapting music to places of interest that the user is visiting (e.g., [4,30]). In the mobile music recommendation system described in [11], the user's location was monitored, music content was analyzed to obtain audio features, and global music popularity trends were inferred from microblogs.

In contrast, the user-related context describes the user's personal state or condition, such as his or her activity, emotional state, and demographic information (e.g., age or gender). The activity information includes an action (e.g., walking, running, driving) or a numerical parameter defining the user's state (e.g., walking pace or heart rate). The user's emotional state cannot be measured directly but is derived from other types of contextual information. In this application type, Elliott and Tomlinson presented a system that adapted music to the user's walking pace by matching the beats-per-minute of music tracks with the user's steps-per-minute [18]. Han et al. and Deng et al. also proposed context-aware music recommendation systems [17,22]. In their works, a music item was recommended according to the user's current emotional state and the music's influence on the user's emotional change.

Some works have combined the above two types (environment and user) of context information for making recommendations. For Example, Baltrunas et al. proposed an approach to perform context-aware music recommendation as the user' drives [5]. The authors took into account different contextual factors (such as driving style, mood, road type, weather, and traffic conditions, gathered via a questionnaire), and these factors were used to extend a matrix factorization model. Also, some works have focused on mobile music consumption; these studies typically match music with the user's current pace while sporting (for Example, [14,43]). To this end, the information about the user's location or heartbeat is used to infer the jogging or walking pace.

The studies most relevant to our work are those making music recommendations based on information collected from smartphones, such as the user's listening behavior, history, or locations [6,51]. The context was obtained from the sensors installed in a mobile device, including the GPS for location and speed, the clock for time, the microphone for the noise level, and RSS feeds for weather and traffic conditions. Recent works include AmbiTune which adapts the music to the drivers based on the prediction of their route trajectories and driving speeds [24]. Such research and surveys have demonstrated that drivers' situations, including their mood and fatigue status, could significantly affect the choice of preferable music while driving [25,26]. In [25], the authors proposed a smartphone-based situation-aware music recommendation system that was designed to turn driving into a safe and enjoyable experience. This system aimed at helping drivers reduce fatigue and negative emotion. It recommended music not only based on drivers' listening behaviors but also on their real-time mood-fatigue levels and traffic conditions. This solution enables different smartphones to collaboratively recommend preferable music to drivers according to each driver's specific situations in an automated and intelligent manner.

As can be observed, previous systems targeting a mobile usage scenario usually consider a single contextual factor directly collected form the wearable sensors. They aim at matching music with the sensor readings that represent the current pace of a walker or jogger [8,30]. These systems typically try to match the user's heartbeat with the music played [40]. However, they often require additional hardware for context logging (e.g., heart rate sensors or pedometers) [14,18,49]. Although using numerous sensors could improve the performance of a recognition algorithm, it is not practical to expect that the users wear all of these in their daily life. Different from the above-mentioned work, we use only the smartphone, without any extraneous equipment, in a natural way to achieve human activity recognition and then use the recognized activities to construct contextual conditions.

In addition to context-aware music recommendation, the other important issue is smartphone-based activity recognition. Many approaches have already been presented in the literature [7,35]. For Example, Kwapisz's work exploited the triaxial accelerometer on a smartphone for HAR [35]. It was able to classify six locomotion activities over intervals of ten seconds. Some researchers performed signal Fourier analysis and a machine learning approach to predict activities of walking, running, cycling and driving with the accelerometer data of a smartphone (e.g., [23,44]). Research on HAR with smartphones is mostly based on accelerometers [36,42]. This is because these embedded inertial sensors were first introduced in the mobile phone market [31].

Many techniques have been proposed to automatically recognize human activities based on different kinds of data [12,39]. They mainly rely on the supervised learning methods [16,41] and differ in the type and the number of sensors, considered activities, adopted learning algorithms and other parameters. The training stage is performed online through the option of collecting the users' live data by following a predefined activity protocol [37]. In addition, ontology-based approaches have been proposed to predict user activities. The defined ontology models a set of activities and context data that can later be used to recognize activities [46]. As can be observed, most of the context-aware music recommendations are data-driven. The researchers often use a set of contextual parameters in a machine learning algorithm (or pre-defined rules) to bind various contexts and music without knowing their associations. An alternative way is to consider a knowledge-driven approach that involves an analytical procedure (or a study deriving knowledge from music psychologists) for understanding what contextual factors might influence music selection and then using such knowledge to construct the relationships between context and music. Different from the above works that recognize only simple human activities, in this study, we adopt and extend machine learning methods to further recognize activities of different levels, and employ different feature selection schemes for performance enhancement. Our work presents a practical approach that separates contextual conditions (i.e., user activities) and user preferences on music. As a result, the recognition results become transparent to the users, and additional knowledge-based rules can be easily included in the mapping between context and music. Computational methods and evaluations are conducted for activity recognition and music recommendation, respectively. The details are described in the following sections.

3. Activity-Aware music streaming recommendation

As mentioned above, this study proposes an activity-aware system for providing music streaming recommendations. In this section, we first present the overall system structure. We then describe the major components of this system and how their corresponding functions are developed in detail.

3.1. System framework

To achieve activity-aware pervasive music recommendations, we present a system framework with a cloud-based, client-server architecture that helps mobile users access the most-suitable on-demand music service. In this work, the client is a smartphone responsible for collecting the user's activity information, recording his feedbacks, and playing streamed music. The reason for using smartphones is that they are currently the most popular handheld mobile devices; other wearable devices can be included for collecting more of the user's contextual information, depending on availability. Meanwhile, the server is constructed on the cloud to manage user profiles, perform signals analysis, train classifiers for activity recognition, and carry out computation for recommendation.

An ideal way to develop an activity-aware system is to construct a single platform that links sensor readings directly to individual music items. That is, the recognized user-activity is used to activate certain music items. However, in reality, it is difficult to collect complete user data (from smartphone signals to music) due to users' concerns over privacy and security. Therefore, this study adopts a compromise where the system is built in two separated phases to achieve the two major functions of activity recognition and music recommendation, and then combines them as the application service. Fig. 1 illustrates our framework, in which the system kernel includes the activity recognition and music recommendation modules. As shown, the two modules are connected by a mapping mechanism that correlates the recognized user activity with a specific music category. The music items belonging to the matched music category are then presented to the user based on the predictive scores obtained by the recommendation module. The above mapping can be implemented in several ways, such as a set of hand-coded rules or a trained classifier. In fact, using the mapping mechanism to connect the recognition and recommendation modules has many advantages. For Example, the activity recognition module can be used to work with other application services, and the learning procedure in this module can be performed on individual users to consider their personal traits. Moreover, the recognition results can be transparent (i.e., why the system recommends these items) and thus interpretable to users.

To detect a specific user activity, the system needs to continuously collect sensor readings from the user's smartphone and then extract features from these time sequence data to train classifiers for recognition afterwards. Based on the recognized results, the system uses various recommendation techniques to produce a song candidate list. The user can activate the service connected to a music streaming site and start playing it. The details of the two modules are described in the following subsections.

3.2. Activity recognition

The aim of activity recognition is to identify the user's actions by unobtrusively observing his behaviors. The activity recognition module in Fig. 2 is organized to be two layers (and can be extended to multiple layers) where activities can be constructed in a hierarchical way. This means that high-level activities are constructed from low-level ones, and the lowest-level activities can be constructed from the raw sensory data. Two inference components are included. The first component is designed for low-level activity recognition. This component is responsible for processing the accelerometer data collected from the smartphone and identifying primitive activities (called micro-activities hereafter). The output of this component indicates the activity inferred. The second component is for high-level activity recognition. It takes the micro-activities as the basic building blocks to identify high-level activities. The recognized result is then used to activate the service requested (i.e., music selection in this study). The above recognition is achieved by machine learning methods operating in the cloud server.

In this activity recognition module, the embedded acceleration sensor is used for data collection. A smartphone has built-in triaxial accelerometers with a 19.6 m/s2 maximum range. In general, the coordinate system of the smartphone is defined relative to the screen with its default orientation. As shown in Fig. 2, the horizontal scale is x-axis, with increasing positive values to the right; the vertical scale is y-axis, with increasing positive values upward; and the dorsoventral scale is z-axis, with increasing positive values outward. In this study, each sample of the accelerometer data is sent

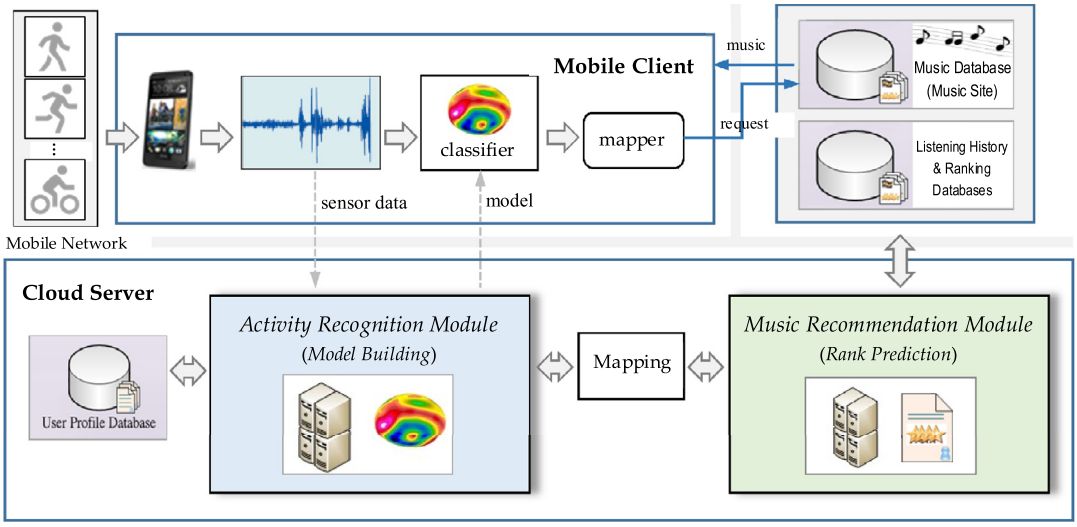

Fig. 1. The system architecture for activity-aware mobile music recommendation.

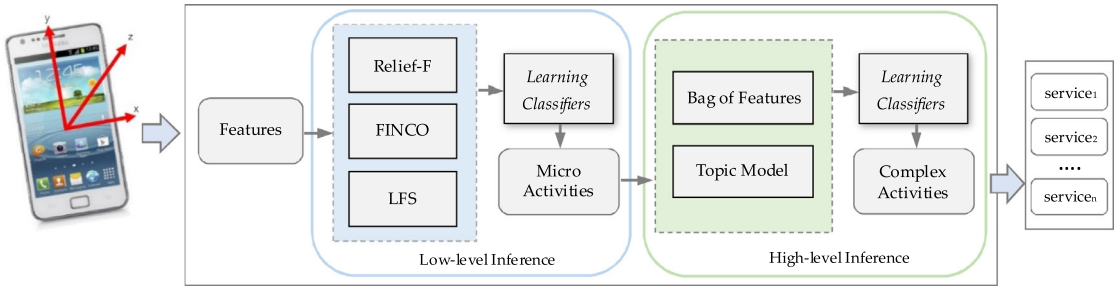

Fig. 2. Schematic diagram of the activity recognition module.

over the network with the format of *x, y, z* values and a timestamp (in nanoseconds).

3.2.1. Micro-Activity recognition

Micro-activities are activities that occur in a short time interval. They are brief and distinct body motions (such as walking) and can be characterized by a statistical sequence of body postures. Here, micro-activities are constituted by triaxial accelerometer readings that are collected within a short period of time, from a few seconds up to minutes. Ten micro-activities are considered, including lying, standing, sitting, walking, jogging, biking, driving, climbing stairs, riding a scooter, and riding an elevator. They are chosen mostly based on the relevant studies, and they cover the motion activities of our daily life. As mentioned above, smartphones with an open-source operating system are chosen as the operating platform here because they are easy to develop, deploy and maintain. To collect data, we create an interactive system to enable physical interaction between users and machines. The open-source smartphone system provides different ways (ranging from the lowest to the highest frequencies) for researchers to collect data from the accelerometer sensors, and we use the operating mode with a frequency of 50 Hz for recording data.

The first phase in activity recognition is to extract specific target features from the time series (raw) data stream. The system can then infer what user activity is taking place. In the feature extraction process, sensor signals are first divided into a number of small time segments (called time windows) with the same time interval. The signals are then transformed into a window-level feature dimension. A non-overlapping technique is used in which each single accelerometer datum belongs to only one time window. The ideal length of the time window is determined by whether the specified interval provides sufficient time to capture several repetitions of motions involved in an activity. For each time window, features are derived from sensor data and are referred to as low-order features. As mentioned above, these low-order features are used to build a classification model. In this study, data are divided into four-second segments, and each segment contains 200 sensor readings. Each raw datum (i.e., reading) is composed of a timestamp, three acceleration values (corresponding to accelerations along the *x*-axis, *y*-axis, and *z*-axis), and two angle values (representing the latitude and longitude).

Two types of low-order features are extracted: time domain and frequency domain features. In accordance with the relevant studies, several time-domain features are used, including the mean, minimum, standard deviation, zero-cross rate, correlation, average resultant acceleration, and average absolute difference. In addition, to extract frequency domain features, we first convert the sensor data into the frequency domain vector and then adopt the most popular frequency domain feature extraction, the Fast Fourier Transform (FFT), to derive features.

Thirty-one informative features (categorized into eleven types) are generated from the 200 raw sensor readings in each time segment. These time and frequency domain features are used to constitute a feature vector for model training; they are listed as below (the value in parentheses indicates the number of features):

Time-Domain Features include:

- Max (3): the maximum acceleration (for each axis);
- Min (3): the minimum acceleration (for each axis);
- Average (3): the average acceleration (for each axis);
- Average Movement Intensity (1): the average of the square roots of the sum of the values of each axis; it is measured

by the following equation, where T is the length of the time window, and $a_x(t)$, $a_y(t)$ and $a_z(t)$ represent the x, y, and z values of the accelerometer data, respectively;

$$AMI = \frac{1}{T}\left(\sum_{t=1}^{T} MI(t)\right), \text{ and}$$
$$MI(t) = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2} \quad (1)$$

- Standard Deviation (3): standard deviation (for each axis);
- Mean Absolute Deviation (3): the average of the absolute values of the sensor deviations;
- Zero Crossing Rate (3): the number of times the signals change the sign in a given period of time; it is measured by the following equation, where x is the time signal of length T, and the indicator function f is equal to 1 if the argument is true (or 0 otherwise);

$$ZCR = \frac{1}{T-1}\sum_{t=1}^{T-1} f(x_t x_{t-1}) < 0 \quad (2)$$

Frequency-Domain Features include:

- Average FFT (3): the average FFT spectrum (for each axis);
- FFT Standard Deviation (3): standard deviation of the spectrum data (for each axis);
- FFT Energy (3): sum of the squared modulus of the coefficients (for each axis); the energy feature captures the intensity of the motion;
- FFT Entropy (3): feature related to the amount of uncertainty about an event associated with a given probability distribution (showing whether the energy is evenly distributed at different frequencies); it is measured by the following equation, where $p(x_t)$ is the probability of X being the state x_t.

$$H(X) = -\sum_{t=1}^{T} p(x_t) log_2 p(x_t) \quad (3)$$

With the above two types of features, a selection scheme is performed to choose a subset of the original features to maximize the performance of a model-learning algorithm. In this way, the dimension of feature vectors required for activity recognition can be reduced, and the computational effort in learning a model (classifier) is thus decreased. In this study, three popular feature selection methods are adopted, including two filter methods, Relief-F and FINCO (Forward and Inconsistency), and one wrapper method, LFS (Linear Forward Selection). They are used to eliminate irrelevant and redundant attributes. The effectiveness of these methods is evaluated. Each feature selection method has its specific characteristics. Relief-F chooses instances randomly and then estimates the relevance of the chosen features according to how well their value distinguishes the data points. The un-important features can be eliminated through the operations. FINCO combines a sequential forward selection with an inconsistency measure. A relevance weight is given to each feature with the goal of selecting the best subset of features. LFS is an attribute selection method with a fixed-set technique for high-dimensional data. It can lower the number of attribute expansions in each forward selection step. More details on the above three algorithms can be found in [2,21,47], and the evaluation results are presented in the experimental section.

3.2.2. Complex activity recognition

As indicated above, our work employs a two-layer approach to recognize complex activities in a hierarchical way. Compared to micro-activities, complex activities involve a relatively longer time that typically lasts a few minutes and even hours. They are structured by a collection of simple activities. That is, simple micro-activities are regarded as primitive components to constitute complex activities. Three complex activities are considered in this study: taking public transportation (transporting), going out for shopping (shopping), and relaxing. The activity of taking public transportation describes the process of a subject taking a public transportation system (for Example, subway, tram or bus). This activity can be constituted by a set of simple activities, such as climbing (stairs), walking, jogging, sitting, standing and driving. The second activity, going out for shopping, means that subjects use their private vehicles, such as bikes, scooters or cars, to go shopping. In a similar way, this activity could be constituted by simple activities, such as biking, driving, riding a scooter, walking, sitting, standing, or riding an elevator. The third activity, relaxing, means that subjects spend their free time talking to their friends, lying in bed, or playing online games. This activity could include simple activities such as lying, sitting, standing, and walking.

Recognition of complex activities can be achieved by first recognizing the micro-activities included in the time period and then classifying the composite sequence of micro-activities involved into an appropriate complex activity. As can be observed, however, the number of possible compositions of simple activities can be very large depending on several factors, such as time and location as well as the person using the device. Therefore, to recognize these activity patterns effectively, a powerful feature extraction mechanism is needed. In this work, two special feature extraction schemes are adopted and evaluated: bag-of-features (BoF) and latent Direchlet allocation (LDA). In the BoF scheme, high-level features are created using histograms of primitive symbols. Here, the BoF records the frequency of each micro-activity (i.e., basic building blocks) appearing in the time window used to describe a specific complex activity. For Example, a complex activity such as relaxing can be defined (represented) as a set of micro-activities, including jogging (0), walking (0), standing (0), climbing stairs (0), biking (0), lying (55), sitting (163), riding a scooter (0), riding an elevator (0), and driving a car (0), where the number in parentheses refers to the frequency of a specific micro-activity detected within a period of fifteen minutes (i.e., a time segment). In this way, the above activity of relaxing is represented as (0, 0, 0, 0, 0, 55, 163, 0, 0, 0).

In contrast, LDA extracts another type of feature, topic features, from topic models that use statistical information from the retrieved data to find a latent structure. This method is often used in the study of information retrieval, and the extracted features are useful in discovering patterns in a large collection of documents by identifying recurring sets of words. LDA is used to model the relationships among words within a topic as well as the correlations among different topics. In this way, each word in each topic can be represented as a probability distribution, and the probability of a word occurring in a document can thus be computed. Here, the words are the labels (names) of the micro-activities, and the document means the labels within a specific time interval. That is, the recognized labels are used as a probabilistic combination for modeling the complex activities. For this case, the Machine Learning for Language Toolkit (Mallet, http://mallet.cs.umass.edu/) is adopted to perform LDA in a Java environment. Similar to the BoF representation, the complex activity is represented as a set of probability distributions. The recognized simple activities are then used as a probabilistic combination for modeling the complex activities.

3.3. Music recommendation

After presenting the activity recognition subsystem, in this section, we show how it can be connected to a streaming music-playing service to make recommendations. Fig. 3 illustrates the functional blocks of this mobile music service, with the core

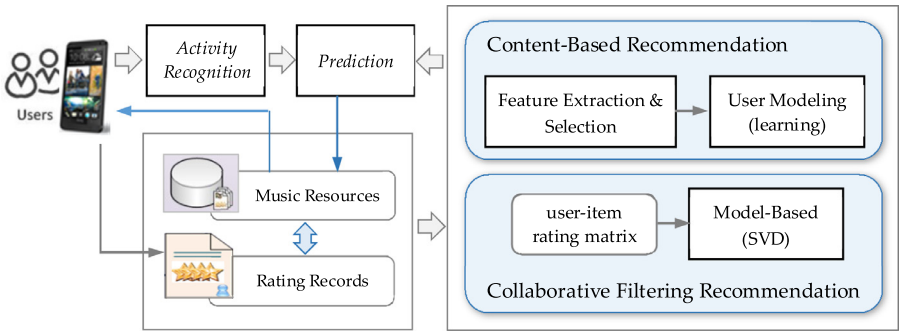

Fig. 3. Schematic diagram of the music recommendation module.

parts of the content-based and collaborative-based recommendation methods. They are used to predict a user's preference of music items in the repository under specific contextual information (here, user activity), and the system can activate the service to retrieve the most-suitable items accordingly.

3.3.1. Content-based recommendation

To perform content-based recommendation, the system needs to create a user profile to serve as a common reference point and then use the obtained information to infer the user's preferences. The items with users' evaluations (ratings) are used as training Examples to build classification models to predict users' preferences on unfamiliar items. Therefore, it is important to collect profile data directly through a questionnaire that gives the user a list of music items and asks for his evaluation (ratings), or indirectly through recording the user's behavior, such as how often he played the music items. It is also important to keep updating the user's profile based on any user-related changes. Here, we adopt the implicit way of recording a user's music playing behavior to intuit his preferences (based on the user's listening-frequencies on music items). The collected data (music items with preference scores) are then used to build user models with a supervised learning method, and the music items are ranked accordingly. Items belonging to the music category matched the recognized user activity are then recommended.

After collecting the evaluation results from the user, the system has to transfer each music item into a symbolic feature form to construct a user's personal model. This involves extracting and modeling semantic information about the music content. In this work, we evaluate three schemes often used in the study of music information retrieval to represent a music item: audio analysis MFCC (Mel-Frequency Cepstral Coefficients), annotation tags, and specific music characteristics, such as genres or singers.

The first scheme means extracting signal-level features by audio analysis. MFCCs are coefficients that collectively make up an MFC, which is a representation of the short-term power spectrum of a sound based on a linear cosine transform of a log power spectrum on a nonlinear melody scale (to indicate that the scale is based on pitch comparisons of frequency). They are derived from a type of cepstral representation of the audio clip (a nonlinear spectrum-of-a-spectrum). In contrast to directly analyzing the audio sequence, which is computationally expensive, the other two schemes use text-based information to capture the semantic information of a music item: they extract specific features to represent a music item. For Example, the tags rock, pop, indie, and vocalist are often used as features to annotate streamed music; and the genres or singers are commonly used as representatives (characteristics) of music. In fact, it is generally agreed that adding annotation metadata to shared content is an efficient way to represent music, and this has been a trend in classifying multimedia items. Many music tagging methods have been proposed to annotate a music item through semantic tags, including auto-tagging (by learning classifiers automatically) and social tagging (performed by a crowd of users manually). To evaluate the performance of the above three schemes, we conduct several sets of experiments (described in Section 4.3) in which all terms appearing in the training Examples are defined as candidate features (attributes) in the learning procedure and are used to construct the user models.

3.3.2. Collaborative recommendations

In addition to the above user modeling method, our work includes a collaborative filtering method that predicts item ratings for a particular user based on how other users previously rated the same items. Here, we adopt a model-based collaborative filtering method, matrix factorization (MF), as it has been considered to deliver better performance than other methods. MF transforms both users and items into the same latent factor space and factorizes the user-item matrix (i.e., the rating matrix) into the product of two matrices. That is, to predict the values of the entries (i.e., the ratings of the un-rated items for users), the rating matrix R is first decomposed into two matrices P (called the P-matrix or user-factor matrix) and Q (called the Q-matrix or item-factor matrix). P and Q are two low-rank matrices (i.e., singular value matrices) corresponding to the user and the item, respectively. This separation occurs so that the two matrices' inner product approximates the original matrix. The goal is to find a decomposition that minimizes the error between the original and the approximation matrices. To predict the rating of an item i by a user u (represented as $r_{u,i}$), we can then calculate their dot product as:

$$r_{u,i} = q_i^T p_u \tag{4}$$

In the above equation, the elements of p_u measure the extent of interest the user u has in items that are high on the corresponding latent factors, and the elements of q_i measure the extent to which the item i possesses those factors. To improve the performance of the above matrix factorization model, researchers have proposed baseline predictors to consider user bias and item bias produced in the rating process [34]. For Example, some users tend to give higher ratings, and certain items tend to receive higher ratings by users. Therefore, in the baseline predictors, the equation for prediction is extended to be:

$$r_{u,i} = \mu + b_u + b_i + q_i^T p_u \tag{5}$$

Where μ is the average rating of all items, b_u is the bias of user u (with respect to μ), and b_i is the bias of item i. To overcome the sparsity problem in rating music items, we develop a new method, SVD+, to consider the user-tagging plus the often used rating. This idea comes from our observation that in some cases, the users do not rate the music items but simply attach tags to them, which indicates that the users are interested in these tagged items. To take into account their tagging information, we add a tagging value to the original item rating: the number of tags a user attached to

this item divided by the total number of tags the user used in this dataset. Then, the newly obtained matrix is used to replace the original rating matrix R in the SVD method. As shown in the experimental results presented in Section 4.3.2, this method can effectively improve the recommendation performance.

In practice, the calculation for rating prediction is often replaced by an iterative learning procedure. The common learning method is the stochastic gradient descent (SGD) algorithm [9], which minimizes the error between the estimated rating and the actual rating. Using the corresponding update rules, we can then perform the operation until the error converges to its minimum. In the SGD algorithm, the true gradient of the objective function (describing the error summed over all training Examples) to be optimized is approximated by a gradient at a single Example at a time. This algorithm continues several iterations over the training set until it converges, and the typical implementation uses an adaptive learning rate to ensure convergence. The error is calculated and used to determine whether the learning procedure must be terminated. Further computational details on the approximation can be found in [34].

4. Experiments and results

After presenting our activity-aware system with strategies in human-machine interaction, information processing and music recommendation, in this section, we describe the system evaluations conducted. As indicated above, our major goal is to demonstrate a practical recommender system. Therefore, we first describe the design and implementation of the system and then the experiments conducted for verification. The aim of the experiments is to evaluate the prediction performance of the user preference under a certain contextual situation; it requires a large collection of complete context-aware data, from sensor signals of a smartphone to the phone-holder's music listening behavior. Nevertheless, the users' concerns regarding privacy and security made it difficult to persuade many of them to install our system on their smartphones for data collection. Considering the aforementioned issues, we employed a compromise method using two separate sets of data for activity recognition and music recommendation, respectively. In the first set of evaluations, participants were asked to tag the activities (user states) in order to train the corresponding classifiers. Then, in the second phase, a dataset collected from a music streaming site was used to evaluate the performance of different methods for music recommendation.

4.1. System design and implementation

A pervasive music recommender system was developed in a cloud-endpoint mobile environment to provide activity-aware music recommendation. It has a client/server architecture divided into two major parts: a user front-end (mobile client) and an application back-end (cloud server). Fig. 4 shows the overall system implementation corresponding to the architecture described in Section 3.1. The system also includes a data storage module and a mechanism for client-server communication, and the database contains tables for raw and intermediate data. In this way, the data visualization, methodology changes, and data insertion can be carried out conveniently. To reduce the overhead of data transmission often required for activity recognition, the data processing mechanism is optimized so that the system can save time waiting for the response from the database. In this work, a set of consecutive sensing records (i.e., 200 time steps) are compressed into a single sample and stored in ARFF format. This process dramatically reduces storage and communication bandwidth requirements. It is especially useful when the data have to be transmitted over the Internet.

Table 1
Statistical comparison of different methods.

	J48	MLP	RF	IBK
Avg	66.4	73	76	83.4
Std	8.6	10.4	9.0	7.9
p-value	$<.05$	$<.05$	$<.05$	–

As described in Section 3.1, the system includes two major modules to achieve activity recognition and music recommendation (as shown in Fig. 5(a)). The micro- activity recognition is performed at the user front-end, where the motion data are collected and the trained model is used to perform classification. Fig. 5(b) shows the interface through which the user can choose to collect and tag different types of activity data for training and to view the data collected. Because model generation and complex activity classification are time-consuming, they are performed at the application back-end (on the cloud server side).

Fig. 6 shows three screenshots (as illustrative Examples) of music recommendation on the mobile client presented to the user. As is shown, the upper-right block provides the detected user activity (e.g., biking), and the upper-left block, the genre of the music recommended (e.g., pop). The play-list with the names of artists and songs is allocated in the middle area of the interface (Fig. 6(a)). Based on the list, the user can start listening to the music (by pressing the blue button, Fig 6(b)) and confirm the recommendation results by giving his feedback to the system (by pressing the orange button on the right, Fig 6(c)). The feedback is then used to update the recommendation mechanism. The music recommendation module is connected to Spotify (www.spotify.com), which gives access to millions of songs. Spotify is probably the best service option for on-demand music streaming at present. It also provides Web API to the users, who can develop the applications themselves to fetch data from the Spotify music catalog and manage their playlists and the saved music. In this way, our system can achieve pervasive and personalized music retrieval and management on mobile devices.

4.2. Performance evaluation for activity recognition

To evaluate the human activity recognition performance, we conducted several sets of experiments. They include the recognition of micro-activities by different classification methods, the evaluation of different feature selection methods used in the classification procedures, and the recognition of complex activities. The experiments are described in the following subsections.

4.2.1. Classification of micro-activities

The first set of experiments aims to evaluate the performance of four popular classification methods on micro-activity recognition: Decision Tree (J48), Multi-layer Perception (MLP), Random Forest (RF) and instance-based k-Nearest Neighbor (IBK). Of the participants invited, nine completed the data collection in this set of experiments. Each participant collected 51 min of sensor data, and there were 765 training instances (with a time window of four seconds) in total. The four methods mentioned above were adopted to build classification models to recognize the micro-activities (as a multi-class classification task), and the strategy of ten-fold cross-validation was used for model testing.

Fig. 7 summarizes the results for the initial activity recognition experiments. It presents the predictive accuracy (averaged over all participants) for each activity by the four learning methods. This figure shows that in most cases IBK achieves a relatively high accuracy. On average, an accuracy of 83.4% for the ten micro-activities is obtained. Table 1 lists the average accuracy and standard deviation for each method, as well as the t-test results (IBK to J48,

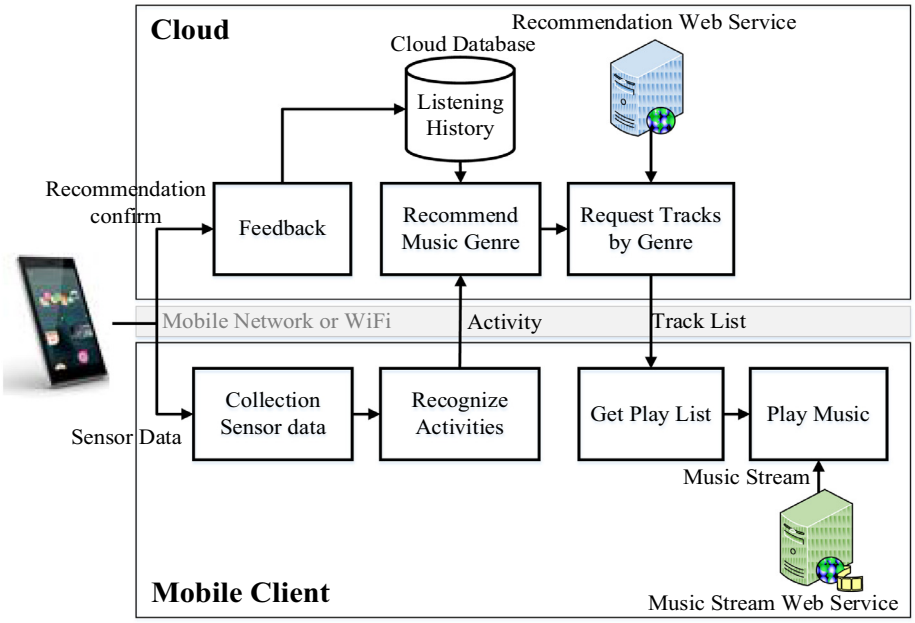

Fig. 4. Implementation of activity-aware music recommendation.

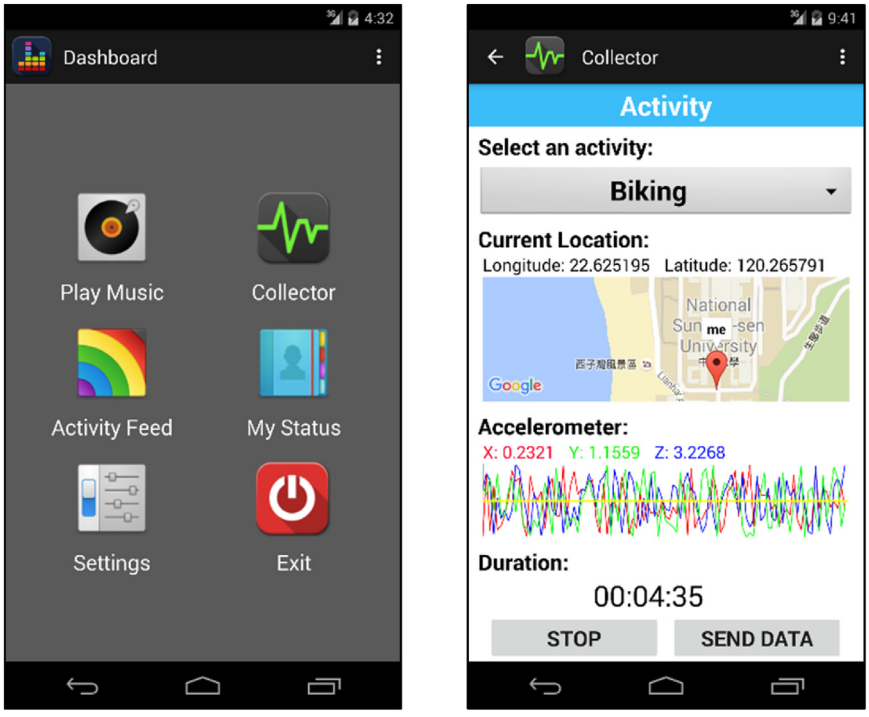

Fig. 5. The initial interface and activity information.

MLP, and RF). As shown, small significant values ($p < 0.05$) obtained from the tests indicate that IBK is statistically better than the other three methods.

The experimental results show that some activities (e.g., sitting) are apparently more easily identified than others, depending on how the acceleration values of the sensor changed. In general, it is expected that the learning models are able to correctly classify different activities, as they involve different forward motion effects in acceleration. However, after examining the misclassified cases, we found that signals (data) of certain activities are similar to each other (such as biking, riding a scooter and driving a car), and this causes misclassifications. For Example, the activity of driving a car was thought to have the most extreme case of forward acceleration changes; however, in real situations, driving a car may have smaller forward acceleration changes than biking when the subject is stuck in a traffic jam and, thus, has slowed his or her pace.

4.2.2. Evaluation of feature selection methods

The aim of the second set of experiments is to investigate how to obtain better results by reducing the computation complex-

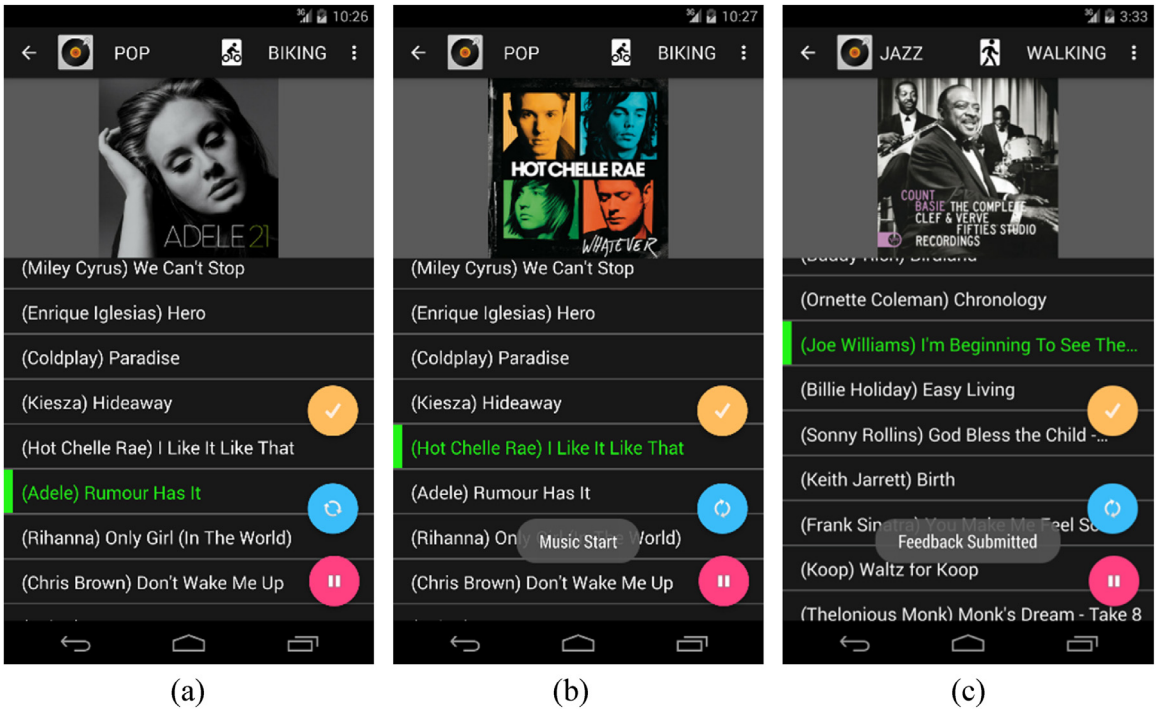

Fig. 6. Screenshots of user interface with music recommendation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article)

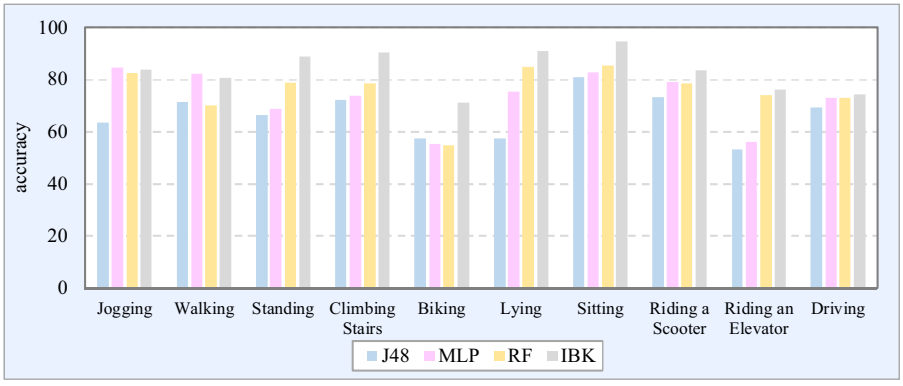

Fig. 7. Recognition results for each activity by four different methods.

Table 2
Features selected by three different feature selection methods.

Method	Time domain features selected	Frequency domain features selected
Relief-F	19 (100%)	12 (100%)
FINCO	8 (42.1%)	6 (50%)
LFS	8 (42.1%)	3 (25%)

ity with an efficient feature selection scheme. In the experiments, three feature selection methods were evaluated: Relief-F, FINCO, and LFS. Based on the experimental results shown in the above section, the IBK method (which gave the best performance) was selected to work with the feature selection methods for evaluation. That is, each feature selection method was applied to the features in the time and frequency domains (as described in Section 3.2), and the original data were reformatted. Then, the IBK method was employed to build classifiers accordingly. The numbers and percentages of the features selected by each method are listed in Table 2. In the table, the second column shows the selected time-domain features and the third column the frequency-domain features.

As shown, among all methods, Relief-F tends to select the greatest number of features. This indicates that Relief-F considers all thirty-one features indispensable for improving the classification accuracy. LFS only selected three features in the frequency domain. This is because some frequency domain features are highly correlated to others, and LFS ignores redundant features.

Theoretically speaking, each feature contributes to the recognition process. However, this does not mean that more features equals better accuracy. To further examine the effect of different numbers of features selected by the three methods, we performed a feature profiling trial: each feature in the candidate feature set was tested by the IBK model. The recognition accuracy under different feature combinations is presented in Fig. 8. These results show that increasing the number of features enhances the recognition performance. It can also be seen that there is no obvious improvement when the number of features selected is increased to a certain extent. For instance, for each feature selection method, there are only slight changes when the number of features selected

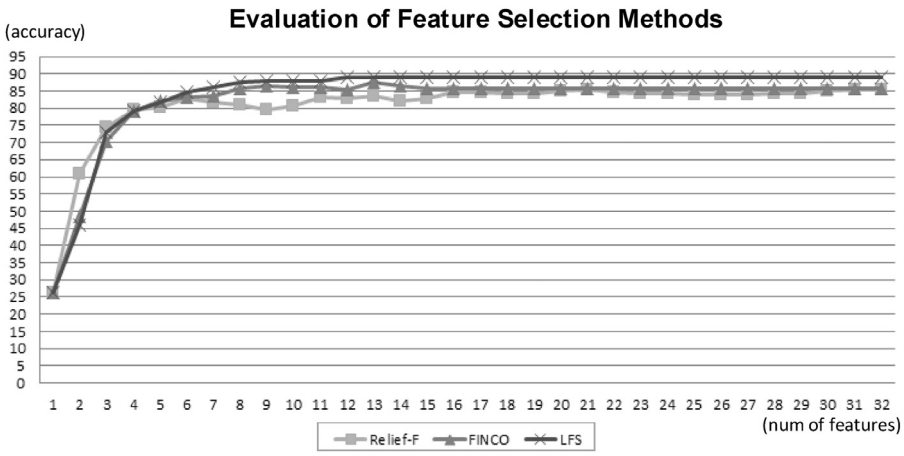

Fig. 8. Testing classification accuracy based on selected features.

Table 3
Confusion matrix for the strategy of using SVM with BoF.

		Predicted Class		
		Transporting	Shopping	Relaxing
Actual class	Transporting	8	0	2
	Shopping	0	4	1
	Relaxing	1	1	11

reaches ten. Finally, the figure shows that the best performance of 89% can be achieved by LFS with 12 selected features. In accordance with the above observations, the model trained by IBK with LFS was adopted in the system to identify the micro-activities appearing within the specified time interval for complex activity recognition.

4.2.3. Classification of complex activities

This set of experiments aims to evaluate the effect of using classification methods with a high-order feature extraction method for complex activity recognition. Several classification methods have been adopted, including Naïve Bayes (NB), Multi-layer Perception (MLP), Random Forest (RF), instance-based *k*-Nearest Neighbor (IBK) and Support Vector Machine (SVM). The two feature extraction methods BoF and LDA were used to work with the above classification methods. For this set of experiments, a total of 572 min of data consisting of 38 training instances were collected to build recognition models for a multi-class classification task.

After the preliminary investigation, we noticed that classification methods with BoF performed much better than with LDA in this application case. The reason could be that LDA often requires more Examples than others to achieve a reasonable effect, while there were relatively few sampling data collected in the time window defined for a complex activity. Therefore, only the results using BoF for feature selection are reported here. The evaluation also shows that none of the above five learning methods consistently outperforms others, although the SVM method is generally a better choice here. Because only a small number of data records were collected and used, we simply list the confusion matrix (Table 3) of the SVM classifier to study and analyze the recognition results (rather than the statistic results often reported for a large dataset). This table indicates that in most cases, the activities can be successfully recognized. After examining the details, we found that the reason why these cases were not recognized correctly is that these complex activities were constituted by the same micro-activities, and the composite activity sequences were similar; therefore, they were difficult to distinguish. More micro-activities are needed to differentiate among the complex activities.

4.3. Performance evaluation for music streaming recommendation

After evaluating the activity recognition module, in this section, we describe a series of trials conducted for mobile music recommendation. Usually, the recommendation experiments involve a large amount of data on users and items. This requires a relatively long collection period. Here, we adopt a dataset from a popular music streaming site that contains registered users and the records of what music they have listened to. Each item is described by an item-id and the corresponding listening frequency. This music site also includes tags specified by different users on music items. In the experiments, two hundreds users with the most-frequent music listening were selected from the database to be the experimental subjects. Because no explicit or implicit preference information was recorded in the original user profiles, we used an indirect method to infer user preference based on the listening frequency of music items. With this strategy, a user u's rating (ranging from 1 to 3) of a music item i is defined as:

$$rating_{u,i} = playcount_{u,i} \times \frac{3}{max_u - min_u} + 1$$

in which $playcount_{u,i}$ indicates how many times this user listened to music item i, and max_u and min_u are respectively the highest and lowest listening frequencies for all music items the user has played on this site. These values were used to predict user preferences by the recommendation methods described below.

4.3.1. Content-Based recommendation

The first set of experiments evaluates the performance of using content-based methods for mobile music recommendation. In the experiments, five different representations were used to describe a music item, including MFCC components, tags, singers, and two hybrid strategies of MFCC with tags and MFCC with singers. There are more possible hybrid MFCC and annotations strategies, but the above two are reported because they give better results than others, according to a preliminary test. Because the MFCC components were not directly available from the music streaming site, we mapped the music items to the other dataset (i.e., Million Song dataset) to extract the relevant information.

To compare the performance of the above representations (i.e., MFCC, tags, singers, and the hybrids), we subjected them to three popular computationally efficient content-based methods: decision tree, support vector machine (SVM), and artificial neural network

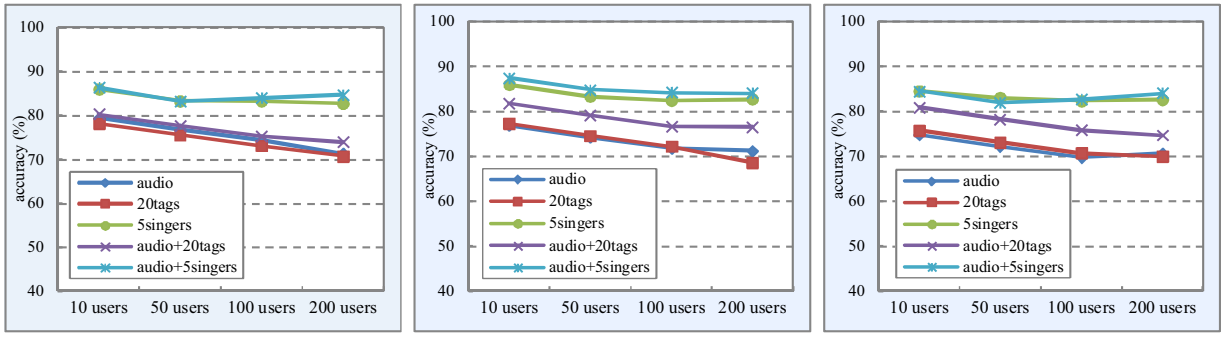

Fig. 9. Results of content-based music recommendation by different methods.

(ANN) classifiers. Moreover, to demonstrate the effect of the number of music items played and rated (implicitly as described above) by the users, different numbers of users were evaluated in the experiments, and those who with more play counts were chosen. The 10-fold cross-validation evaluation method was employed to obtain a more objective assessment.

Since this set of experiments aimed to observe the corresponding effects and tendency of different strategies in user modeling (rather than to determine the best method by an extensive investigation, as the performance may depend on the dataset used), we simply present the accuracy in music prediction. Fig. 9 shows the results, in which values are averaged over the number of users indicated in the *x*-axis. As can be observed, the singer-based method outperforms the audio-based (MFCC) and tag-based methods, and the hybrid singer and audio strategy obtains the best results. One possible reason for this could be that with this music streaming site, returning users often have their favorite singers in mind and play their music accordingly. As a result, the singer becomes the most dominant factor, and it can more pertinently capture user preferences in music recommendations. It can also be seen that in general, the above three machine-learning techniques can deliver reasonably good performance and are ideal for user modeling. In addition, these figures show that the performance declined when more users were considered in the experiments. This is because users were chosen based on their play counts (in decreasing order); the users recently included were those with relatively small play counts due to insufficient learning Examples, and this situation deteriorated the prediction performance.

4.3.2. Collaborative recommendation

In addition to the content-based strategy, we conducted another set of experiments to examine the effectiveness of the collaborative filtering methods. In these experiments, three collaborative strategies were used: the SVD, the enhanced SVD methods (called SVD+) and the *k*-NN method. The corresponding results are compared. For the *k*-NN method, we first measured the similarity among users and specified a threshold to ensure minimal similarity between a selected neighbor and the user. In the results reported below, parameter *k* was 5 and the similarity threshold was 0.8, based on a preliminary test. The number of users was 500 in this set of experiments.

Fig. 10 presents the test results of the collaborative methods, with the representative results of the content-based methods collected from the above section shown for comparison. In addition to the predictive accuracy, the RMSE (root mean squared error) was measured, which is an important performance criterion often used to evaluate collaborative filtering methods. Moreover, it has been emphasized and demonstrated that in collaborative filtering studies that even small improvements in RMSE are considered valuable within the context of recommender systems, as indicated in [19]. As can be seen in the figure, by exploiting the advantages

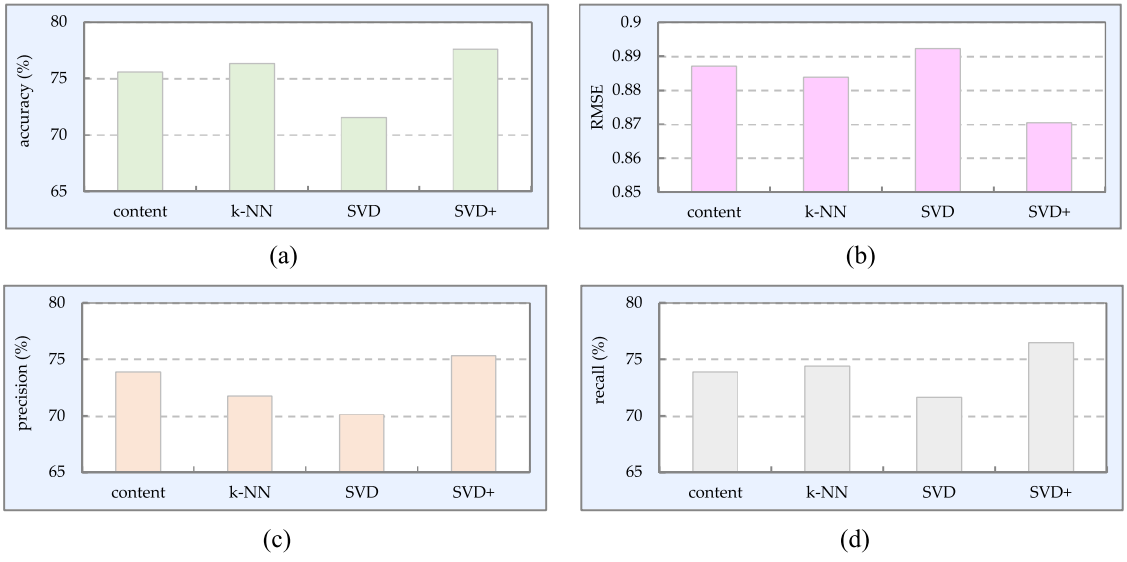

Fig. 10. Results of collaborative recommendation.

of grouping people in the same dataset, the collaborative filtering methods *k*-NN and SVD+ performed better than the content-based method. In particular, SVD+ can give the best results. From this figure, we can also observe that the content-based method (i.e., the best result of Fig. 9) provides better recommendations than the traditional SVD method. This is because of the data characteristics of this music streaming site, in which users focus on their favorite singers, as already analyzed in the above section. The performances of two more evaluation metrics, precision and recall are provided in Fig. 10(c) and Fig. 10(d), respectively. These figures show that results consistent with the prediction accuracy and RMSE mentioned above were obtained.

Although the results presented in Fig. 10 indicate that the enhanced model-based SVD+ method performed best in this series of experiments, it also has some drawbacks. For Example, the latent factor models often employ a stochastic algorithm to solve the optimization task. In our observations, when the number of users and items increases, the corresponding optimization task becomes more difficult to solve. The cold start and data sparsity problems often associated with collaborative filtering methods also require more attention. All methods have advantages, and the choice mainly depends on the specific service required.

5. Conclusions

Over the past few years, increasingly more sensors have been embedded into smartphones to track and collect different types of context information about users and their environments. Human activity is a special kind of contextual information and can be combined with the perceived environmental data to determine appropriate service actions. In this study, we developed a smartphone-based activity-aware system that infers human activities to make appropriate music streaming recommendations. Our work adopted machine learning algorithms to recognize user activities and to estimate user preferences of songs in various situations. Different methods for extracting time and frequency domain features from smartphone signals were evaluated. Comparisons were made of methods coupling different music representation schemes with recommendation techniques.

Our work presents a practical approach to separating contextual conditions and user preferences regarding music. In this way, the recognition results can be made transparent to the users and additional knowledge-based rules can be easily included in the mapping between user contexts and music items. Several sets of experiments were conducted for performance evaluation. The results show that the accuracy of different methods varies from 66.4% to 83.4% in activity recognition, and from 71.5% to 77.6% in music recommendation. In addition, the results were analyzed from different perspectives. All confirmed the feasibility and effectiveness of the approach developed herein. These results show the promise and potential of the presented approach. In addition, to undertake the proposed activity-aware approach, we implemented a mobile music recommendation system on a cloud platform to demonstrate how we tailored user activities as contextual information with mobile devices to develop a real-world application.

Based on the presented work, we are currently investigating new ways to improve the system. The experiments in this work were conducted under various restrictions and required a compromise solution for data collection. To overcome this difficulty, we plan to collect complete user information to enrich the dataset through an on-line questionnaire interface for linking smartphone data directly to music items with user preferences. In addition, we are investigating the possibility of cooperating with a local telecommunications service provider to develop a practical method of collecting user data in the real world. Meanwhile, we have been integrating more sensor signals from wearable devices with the proposed framework to recognize more wide-ranging human activities in our daily life and to provide more-precise mobile music recommendations.

References

[1] E.P. Abril, Tracking myself: assessing the contribution of mobile technologies for self-trackers of weight, diet, or exercise, J. Health Commun. 21 (6) (2016) 638–646.

[2] E. Acuna, A comparison of filters and wrappers for feature selection in supervised classification, in: Proceedings of the Thirty-Fifth Symposium on the Interface of Computing Science and Statistics, 2003, pp. 613–625.

[3] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, in: F. Ricci, L. Rokach, B. Shapira, P. Kantor (Eds.), Recommender Systems Handbook, 2011, pp. 217–253.

[4] A. Ankolekar, T. Sandholm, Foxtrot: a soundtrack for where you are, in: Proceedings of Interacting with Sound Workshop: Exploring Context-Aware, Local and Social Audio Applications, 2011, pp. 26–31.

[5] L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Lüke, R. Schwaiger, Incarmusic: contextaware music recommendations in a car, in: Proceedings of the Twelfth International Conference on E-Commerce and Web Technologies, 2011, pp. 89–100.

[6] L. Baltrunas, B. Ludwig, S. Peer, F. Ricci, Context relevance assessment and exploitation in mobile recommender systems, Pers. Ubiquitous Comput. 16 (5) (2013) 507–526.

[7] M. Berchtold, M. Budde, D. Gordon, H.R. Schmidtke, M. Beigl, Activity recognition service for mobile phones, in: Proceedings of International Symposium on Wearable Computers, 2010, pp. 1–8.

[8] J.T. Biehl, P.D. Adamczyk, B.P. Bailey, DJogger: a mobile dynamic music device, in: Proceedings of International Conference on Human Factors in Computing Systems, 2006, pp. 556–561.

[9] L. Bottou, O. Bousquet, The tradeoffs of large scale learning, Adv. Neural Inf. Process. Syst. 20 (2008) 161–168.

[10] F. Cena, S. Likavec, I. Lombardi, C. Picardi, Should I stay or should I go? improving event recommendation in the social web, Interacting Comput. 28 (1) (2016) 55–72.

[11] Z. Cheng, J. Shen, On effective location-aware music recommendation, ACM Trans. Inf. Syst. 34 (2) (2016) article no. 13.

[12] D.J. Cook, N.C. Krishnan, P. Rashidi, Activity discovery and activity recognition: new partnership, IEEE Trans. Cybernetics 43 (3) (2013) 820–828.

[13] M. Conti, S.K. Das, C. Bisdikian, M. Kumarb, L.M. Ni, A. Passarella, G. Roussos, G. Tröster, G. Tsudik, F. Zambonelli, Looking ahead in pervasive computing: challenges and opportunities in the era of cyber-physical convergence, Pervasive Mobile Comput. 8 (2012) 2–21.

[14] S. Cunningham, S. Caulder, V. Grout, Saturday night or fever? Context-aware music playlists, in: Proceedings of the Third International Audio Mostly Conference of Sound in Motion, 2008.

[15] Y. Dash, S. Kumar, V.K. Patle, et al., A novel data mining scheme for smartphone activity recognition by accelerometer sensor, in: S. Das, et al. (Eds.), Advances in Intelligent Systems and Computing, 2015, pp. 131–140.

[16] W.-Y. Deng, Q.-H. Zheng, Z.-M. Wang, Cross-person activity recognition using reduced kernel extreme learning machine, Neural Netw. 53 (2014) 1–7.

[17] S. Deng, D. Wang, X. Li, G. Xu, Exploring user emotion in microblogs for music recommendation, Expert Syst. Appl. 42 (23) (2015) 9284–9293.

[18] G.T. Elliott, B. Tomlinson, PersonalSoundtrack: context-aware playlists that adapt to user pace, in: Proceedings of ACM Conference on Human Factors in Computing Systems, 2016, pp. 736–741.

[19] R. Forsati, M. Mahdavi, M. Shamsfard, M. Sarwat, Matrix factorization with explicit trust and distrust side information for improved social recommendation, ACM Trans. Inf Syst. 32 (4) (2014) article no. 17.

[20] M. Gillhofer, M. Schedl, Iron maiden while jogging, debussy for dinner? an analysis of music listening behavior in context, in: *Proceedings of MultiMedia Modeling*, Lecture Notes in Computer Science 8936, 2015, pp. 380–391.

[21] M. Gutlein, E. Frank, M. Hall, A. Karwath, Large-scale attribute selection using wrappers, in: Proceedings of IEEE Symposium on Computational Intelligence and Data Mining, 2009, pp. 332–339.

[22] B. Han, S. Rho, S. Jun, E. Hwang, Music emotion classification and context-based music recommendation, Multimedia Tools Appl. 47 (2010) 433–460.

[23] Z. He, L. Jin, Activity recognition from acceleration data based on discrete consine transform and SVM, in: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 2009, pp. 5041–5044.

[24] P. Helmholz, S. Vetter, S. Robra-Bissantz, AmbiTune: bringing context-awareness to music playlists while driving, in: *Advancing the Impact of Design Science: Moving from Theory to Practice*, Lecture Notes in Computer Science 8463, 2014, pp. 393–397.

[25] X. Hu, J. Deng, J. Zhao, W. Hu, et al., SAfeDJ: a crowd-cloud codesign approach to situation-aware music delivery for drivers, ACM Trans. Multimedia Comput., Commun. Appl. 12 (1) (2015) article no. 21.

[26] P.G. Hunter, E.G. Schellenberg, A.T. Griffith, Misery loves company: mood-congruent emotional responding to music, Emotion 11 (5) (2011) 1068–1072.

[27] K. Ji, R. Sun, W. Shu, X. Li, Next-song recommendation with temporal dynamics, Knowl.-Based Syst. 88 (2015) 134–143.

[28] A. Kailas, C.C. Chong, F. Watanabe, From mobile phones to personal wellness dashboards, IEEE Pulse 1 (1) (2010) 57–63.

[29] M. Kaminskas, F. Ricci, Contextual music information retrieval and recommendation: state of the art and challenges, Comput. Sci. Rev. 6 (2012) 89–119.
[30] M. Kaminskas, F. Ricci, M. Schedl, Location-aware music recommendation using auto-tagging and hybrid matching, in: Proceedings of ACM Conference on Recommender systems, 2013, pp. 17–24.
[31] W.Z. Khan, Y. Xiang, M.Y. Aalsalem, Q. Arshad, Mobile phone sensing systems: a survey, IEEE Commun. Surv. Tutorials 15 (1) (2013) 402–427.
[32] Y. Kim, H. Shin, Y. Chon, H. Cha, Crowdsensing-based Wi-Fi radio map management using a lightweight site survey, Comput. Commun. 60 (2015) 86–96.
[33] P. Knees, M. Schedl, A survey of music similarity and recommendation from music context data, ACM Trans. Multimedia Comput., Commun. Appl. 10 (1) (2013) article no. 2.
[34] Y. Koren, R. Bell, Advances in collaborative filtering, in: F. Ricci, L. Rokach, B. Shapira, P. Kantor (Eds.), Recommender Systems Handbook, 2011, pp. 1–42.
[35] J.R. Kwapisz, G.M. Weiss, S.A. Moore, Activity recognition using cell phone accelerometers, SIGKDD Explorations Newslett. 12 (2011) 74–82.
[36] O. Lara, M. Labrador, A survey on human activity recognition using wearable sensors, IEEE Commun. Surv. Tutorials 1 (2012) 1–18.
[37] O.D. Lara, A.J. Pérez, M.A. Labrador, J.D. Posada, Centinela: a human activity recognition system based on acceleration and vital sign data, Pervasive Mobile Comput. 8 (2012) 717–729.
[38] W.-P. Lee, K.-H. Lee, Making smartphone service recommendation by predicting users' intentions: a context-aware approach, Inf. Sci. 277 (2014) 21–35.
[39] O.D. Lara, M.A. Labrador, A survey of human activity recognition using wearable sensors, IEEE Commun. Surv. Tutorials 15 (3) (2013) 1192–1209.
[40] H. Liu, J. Hu, M. Rauterberg, Follow your heart: heart rate controlled music recommendation for low stress air travel, Interact. Stud. 16 (2) (2015) 303–339.
[41] Y. Liu, L. Nie, L. Han, L. Zhang, D.S. Rosenblum, Action2Activity: recognizing complex activities from sensors, in: Proceedings of International Joint Conference on Artificial Intelligence, 2015, pp. 1617–1623.
[42] A. Mannini, A.M. Sabatini, Machine learning methods for classifying human physical activity from on body accelerometers, Sensors 10 (2010) 1154–1175.
[43] B. Moens, L. van Noorden, M. Leman, D-Jogger: syncing music with walking, in: Proceedings of the Seventh Sound and Music Computing Conference, 2010, pp. 451–456.
[44] T. Plötz, N.Y. Hammerla, P. Olivier, Feature learning for activity recognition in ubiquitous computing, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, 2011, pp. 1729–1734.
[45] D. Querciaxy, N. Lathiaz, F. Calabresey, G. Di Lorenzoy, J. Crowcroftx, Recommending social events from mobile phone location data, in: Proceedings of IEEE International Conference on Data Mining, 2010, pp. 971–976.
[46] D. Riboni, C. Bettini, COSAR: hybrid reasoning for context-aware activity recognition, Pers. Ubiquitous Comput. 15 (2011) 271–289.
[47] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, Mach. Learn. 53 (2003) 23–69.
[48] M. Schedl, A. Flexer, J. Urbano, The neglected user in music information retrieval research, J. Intell. Inf. Syst. 41 (2013) 523–539.
[49] M. Schedl, G. Breitschopf, B. Ionescu, Mobile music genius: Reggae at the beach, metal on a Friday night? in: Proceedings of International Conference on Multimedia Retrieval, 2014, pp. 507–510.
[50] M. Strobbe, O. van Laere, F. Ongenae, S. Dauwe, B. Dhoedt, F. de Turck, P. Demeester, K. Luyten, Novel applications integrate location and context information, IEEE Pervasive Comput. 11 (2) (2012) 64–73.
[51] X. Wang, D. Rosenblum, Y. Wang, Context-aware mobile music recommendation for daily activities, in: Proceedings of ACM International Conference on Multimedia, 2012, pp. 99–108.