J Internet Serv Appl (2012) 3:195–214
DOI 10.1007/s13174-012-0061-3

ORIGINAL PAPER

A proactive personalised mobile recommendation system using analytic hierarchy process and Bayesian network

Kam Fung Yeung · Yanyan Yang · David Ndzi

Received: 8 November 2010 / Accepted: 26 May 2012 / Published online: 20 July 2012

Abstract With the growth, ready availability and affordability of wireless technologies, proactive context-aware recommendations are a potential solution to overcome the information overload and the common limitations of mobile devices (inconvenience of data input and Internet browsing). The automatic provision of just-in-time information or recommendations tailored to each user's needs/preferences contextualised from the user's activities, location, usage patterns, time, and connectivity may not only facilitate access to information but also remove barriers to the adoption of current and future services on mobile devices. This paper describes a hybrid P2P context-aware framework called JHPeer which supports a variety of context-aware applications in mobile environments. Any context-aware information services such as recommendation services could use the collected and shared contextual information in JHPeer network. An analytic hierarchy process based multi-criteria ranking (AHP-MCR) approach has been developed and used to rate recommendations in a variety of domains. The weights of the contexts criteria can be assigned by the user or automatically adjusted via individual-based and/or group-based assignment. Additionally, a Bayesian network algorithm is applied to solve the cold-start problem in recommendation systems. The paper also proposes a strategy for using Bayesian networks for recommendation services. A news recommendation application has been implemented on the developed JHPeer framework, which proactively pushes relevant news based on the users' contextual information. Evaluation studies show that the system can push relevant recommendations to mobile users appropriately.

Keywords Proactiveness · Information push · Mobile recommender system · Personalised recommendation · Bayesian network · Analytic hierarchy process · Content-based filtering · Collaborative filtering

1 Introduction

Mobile information recommendation is becoming very popular due to the growing diversity, availability and use of mobile information services. A key challenge is to make information relevant within the mobile user's dynamic environment, and to automatically push relevant and interesting information to mobile users.

Many studies have been conducted on the provision of context-aware information services such as personalised news delivery [1–4]. However, most systems are user driven and require users to express their interests and input the keywords. In mobile environment, due to the limitations of mobile devices, data input and information browsing are inconvenient. A desirable solution is to proactively deliver relevant information to the user's mobile device. A mobile system can keep track of a user's activities, preferences, actions, locations, time and connectivity. Automatic user profiling and information recommendation using these context information have the potential to deliver more relevant and just-in-time information to mobile users with reduced keystrokes.

Most of the previous recommendation systems [1,2,5,6] share two major limitations. Firstly, each system is designed and built for a specific type of context-aware

K. F. Yeung (✉) · Y. Yang · D. Ndzi
School of Engineering, University of Portsmouth,
Anglesea Building, Anglesea Road, Portsmouth, PO1 3DJ, UK
e-mail: kam.yeung@port.ac.uk

Y. Yang
e-mail: linda.yang@port.ac.uk

D. Ndzi
e-mail: david.ndzi@port.ac.uk

recommendation such as movie, activity, etc., which limits the extensibility of the recommender to other application domains. Secondly, context-aware recommendation systems incorporate similar contextual information such as time, location, user's interest, etc. Previous studies have organised the collected context data in various structures which are not possible to share between context-aware services.

To resolve these problems, this paper presents a generic hybrid peer-to-peer (P2P) context-aware framework called JHPeer (JXTA based Hybrid peer-to-peer)[7,36]. JHPeer is a generic framework which supports rapid development of context-aware applications and enables context information sharing and dissemination across different applications in mobile environments. It supports proactive information pushing rather than the traditional push-pull methods. Besides, JHPeer framework provides a set of pre-defined contextual documents and is able to store and transmit users' low-level contextual information between peers. Context-aware services based on JHPeer framework can re-use collected contextual information.

Ranking process is one of the vital processes in recommendation systems. A number of methods have proposed for recommendation system such as content-based filtering (CBF) [8], collaborative filtering (CF) [9] and hybrid approaches [10]. These approaches mostly utilise user profiles in the ranking process and they usually consider a single-criterion value (i.e., overall rating). However, users may depend on more than one utility aspect when making their choices. For instance, Yahoo! Movies! (http://movies.yahoo.com/) have employed multi-criteria ratings. Rather than an overall rating, it provides four additional ratings (i.e., Story, Acting, Direction, Visuals). A number of studies investigated the use of multi-criteria approach [11–13] in order to incorporate additional contextual information (criteria) such as time, gender, age, etc. Nevertheless, these studies mainly targeted web-based environment and neglected dynamic contexts and did not model the dynamic changes in users' interests.

To address these limitations, the authors have proposed an Analytic Hierarchy Process (AHP) based on Multi-Criteria Ranking (AHP-MCR) approach to handle dynamic real-time contextual data in ranking process. Rather than developing different AHP models for different domains, we summarise a general AHP hierarchy model through empirical studies. AHP-MCR approach provides the flexibility to add/remove contextual criteria in different scenarios and domains. (Examples are shown in Sect. 4.2.1). It also enables user adjustment of criteria weights based on individual user's needs. For instance, a user may want to read recent news or only view popular news.

Pushing is another crucial process for a proactive recommender. A traditional recommender simply recommends top-k items to users based on users' requests. In a ubiquitous mobile recommendation system, it is more desirable to proactively push just-in-time information to users than to require users to search for information. This paper identifies four types of approach to determine when/how to push recommendations to mobile user with minimum intrusiveness.

Moreover, cold-start problem still exists in multi-criteria recommender. This paper presents a strategy to apply Bayesian network (BN) to eliminate the problem of data sparsity for new users.

This paper makes the following contributions. Firstly, it presents a generic context-aware framework in hybrid P2P environment and discusses some of the potential application domains. Secondly, it proposes an AHP-MCR approach to dealing with the dynamic context criteria problem. Thirdly, it develops a general AHP hierarchy model for contextual recommender through empirical studies. Fourthly, it presents a strategy of using BN to deal with cold-start problem. Fifthly, it implements a proactive context-aware news recommender on top of the JHPeer framework. Finally, it evaluates the effectiveness and the efficiency of the proposed JHPeer framework and the AHP-MCR approach through experiments.

This paper is organised as follows: Sect. 2 summarises related studies in context-aware recommendation systems. Section 3 presents the JXTA-based hybrid peer-to-peer framework, details its mechanism and implementation layers. Section 4 introduces the AHP-MCR approach. Section 5 describes the Bayesian network used to eliminate data sparsity for new users. Section 6 presents the implementation of PPNews and details the algorithms used. Experimental results are presented in Sect. 7. Finally, conclusions and discussions of possible extensions to the system are provided in Sect. 8.

2 Background and related work

2.1 Personalised recommendation system

One of the earliest recommendation system is Tapestry [14] which filters e-mails using collaborative filtering. A significant number of studies have been undertaken on providing personalised recommendation to users in many different domains such as news [15], movie [16], television [17], books [18], products [19], activity [1], music [20], tourism [21], etc. These recommendation systems could be generally categorised into two types [11,22,23]: content-based filtering (CBF) [8] and collaborative filtering (CF) [9]. Content-based filtering generates a user profile from the user history and compares it with new items to make recommendations. Collaborative filtering generates recommendations by analysing community users' histories and ratings. However, the main drawback of both methods is the cold-start [9,23,24] problem where they may not be able to provide relevant recommendations for new users or rate new items.

To overcome this problem, a number of studies have been proposed to combine several techniques to form a hybrid recommendation system [9,15,24]. However, these systems are mainly designed for traditional web recommendations and do not put users' contextual information into contexts.

2.2 Context-aware recommender systems

Context information provides an important basis for understanding mobile users' information needs. With the development of technologies and mobile devices, recommendation systems can incorporate users' ambient contextual information (such as users' activities, preferences, actions, locations, weather, time, with whom, etc.) to deliver more relevant recommendations to users. There have been a number of context-aware recommendation systems [2,3,8,12,16,25–27] that use contextual attributes and combine implicit/explicit user preferences in making recommendations. News@hand [8] adopted semantic context-aware technologies to combine content-based filtering and collaborative filtering. Chen [25] proposed a context-aware CF system that used context similarity techniques to recommend items. A user who has a similar context to previous user would be considered to have similar interests. Yap [26] proposed a context-aware recommender that was combined with an implicit user feedback technique. Ono [16] presented a context-aware movie recommender that took preference and situation (whom, where, mood) contexts into account in its filtering process.

Recent recommenders consider users' contextual information, but not many recommendation systems have the ability to analyse, capture and share contextual information from mobile devices. In order to manage contextual information in context-aware systems effectively, a context-aware framework has to be involved. Basically, a context-aware framework consists of several components such as system/network infrastructure, context retrieval/aggregator, context model, context processing/reasoning and resource/context discovery [28]. Using these components, it is able to provide personalised services.

2.3 Peer-to-peer based recommendations

Manouselis [11] has suggested that modern recommender systems should be based on peer-to-peer (P2P) architecture, taking advantage of the distributed nature of the system. A number of P2P based recommendation systems have been proposed. DYNAMOS [27] demonstrates various mobile services based on a context-aware system in a hybrid P2P network, which automatically delivers service descriptions to mobile users for subscription purposes. It supports client-to-client communication in addition to client-to-server communication for information sharing. A mobile user can subscribe to a weather service to receive real-time weather information or subscribe to nearby user nodes to receive user comments. However, it is designed to run on specific mobile platform (i.e., Symbian OS) and does not provide a context model to hold contextual information. It has the incompatibility issue in the heterogeneous mobile environments. The context data from different devices and platforms is not exactly the same. CARS [3] proposed a service-oriented P2P environment that makes use of discovery function and nearest-neighbours algorithm for collecting user contexts from peers in order to generate recommendations. However this research has not been implemented and evaluated. DeHinter [29] presented a pure P2P recommender in which users were self-clustered in the peer network. It introduced the term proactive recommendation in P2P architecture. However, a pure P2P architecture is not suitable for mobile devices since it requires mobile peers to handle numerous queries from other peers.

The client–server based approaches suffer from a single point of failure and have limited scalability while the P2P based approaches may overload the network and be less efficient. This research develops a context-aware framework based on hybrid P2P technologies to provide personalised services in heterogeneous mobile environment. It enables the deployment of multiple servers to implement distributed services and perform critical load balancing. Besides, by making use of contextual information of mobile devices, the recommendation service can generate user profiles automatically. Meanwhile, its information pushing function can offer a proactive recommendation service to users.

2.4 Proactive information push

The key technology for proactive recommendations is automatic user profiling and filtering. The proactive recommender should sense the user's environment/situation and autonomously [30] predict the user's needs and deliver the relevant contents at the right time [31].

Proactive mobile recommendation systems have been implemented in some of the previous research work [27,32,33]. NAMA [32] uses a proactive reminder system based on associative theory by making use of user contextual information such as profile, preference and location. However, it does not take group contexts/heuristic contexts into consideration. It is also not user friendly since it repeatedly alerts the user until the user switches to pull-based service mode. DYNAMOS [27] provides context-aware service that operates in a hybrid peer-to-peer fashion, but it does not consider users' information needs. Once a user's context is matched, it pushes the service descriptions. Choeh [33] implements a personalised recommender that pushes items based on defined rules. However, these approaches are mainly policy-based that do not take users' history and situation context into consideration, which may result in a bad user experience.

2.5 Multi-criteria recommendation approach

Multi-criteria decision making (MCDM) methods have recently become popular [2,11,23] for implementing recommender systems that use multiple criteria rather than traditional single criteria. The traditional recommender rates items based on (R: users × items → ratings), while MCDM may involve (R: users × items × contexts × rates × time → ratings).

Using a multi-criteria approach, Adomavicius [12] has presented a general approach (multidimensional model) on how to handle contextual information in ranking. Manouselis [34] carried out experiments to compare the performance of MCDM with that of CF algorithms. Zhang [13] applied Probabilistic Latent Semantic Analysis (PLSA) to MCDM recommender using Yahoo! Movie dataset and showed that it improved the performance in recommending top-k items than the single criterion. However, most of the research developments were targeted at desktop/web recommendations and did not focus on mobile environment. Besides, most of the researches relied upon the computation of the weights of contexts using simple weighted sum – SUM (context × weight). Nevertheless, we utilise MCDM methods and develop a general ranking model based on AHP for context-aware recommender. The weight of each context (criterion) is assigned by the user or automatically adjusted via individual-based and/or group-based (group decision making) assignment. The AHP criteria hierarchy is extendable by adding sub hierarchies (sub-criteria). Furthermore, any rating-based CF/CBF algorithms can be integrated in the ranking process.

For mobile users, Kwon [35] presents a context-aware message filtering system that uses case-based reasoning and AHP. However, it is limited by the number of cases in the database. Besides, the author used a user questionnaire to assign the importance of all criteria. This method is inflexible if the number of criteria increases because it would require a long questionnaire to be created.

3 JHPeer framework

Currently context-aware applications in mobile environment can be developed using low-level APIs supported by modern mobile devices. The developers have to focus too much on the technical aspects and not the services aspects. The key principle of JHPeer is to provide the necessary high-level concepts and APIs for rapid development of various context-aware applications for mobile users.

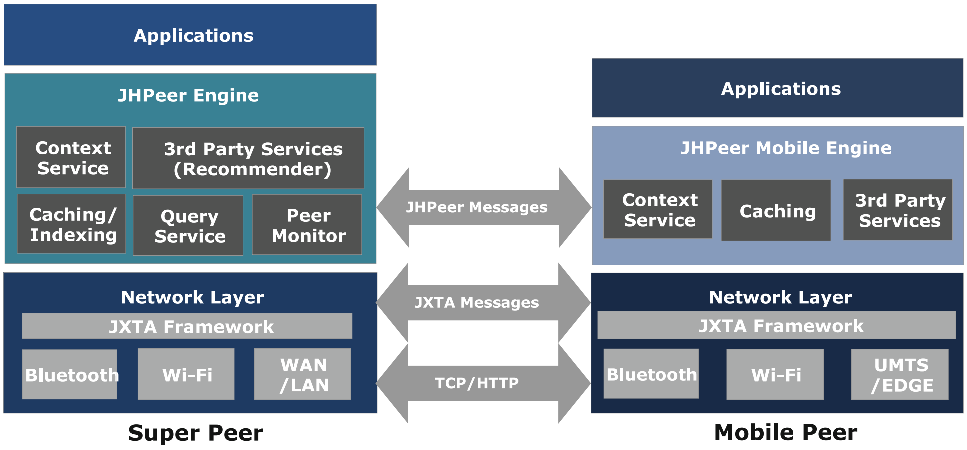
Figure 1 shows the layered structure of JHPeer framework focusing on networking, mechanism and application respectively. This separation of layers not only allows their independent extension, but also simplifies system design by isolating and addressing system properties at appropriate levels of abstraction.

As depicted in the figure, The JHPeer framework is divided into three layers:

Network layer: At the lowest level, the layer encapsulates the implementation details such as interconnection networks, network specific APIs. This layer contains various implementations for persistent connection amongst peers and connecting peers via various media.

JHPeer (mobile) engine: In the centre of the framework, this layer provides abstractions of peer and context data management to support various application domains. The JHPeer engine consists of five key components: (1) Context Service that handles context-related operations such as context discovery, acquisition, analysis, and distribution. (2) Peer Monitor that checks and reports current peer status periodically to nearby peers in order to indicate active mobile devices. (3) Query Service that performs distributed discovery functions for retrieving indexed documents in the peer network. (4) Caching/Indexing that stores, manages, and retrieves documents in local storage and builds index files to enable fast searching. (5) 3rd Party Services that enable developers to implement application specific components such as data replication service, information recommendation services.

Fig. 1 JHPeer framework



The key feature of this layer is reusability and composability. Section 3.1 gives a detailed technical description of the main components in JHPeer engine layer.

Application layer: various applications use the interface provided by the framework's core functionality for its specific needs.

JHPeer consists of two types of peers: mobile peer and super peer.

Mobile peers are usually the mobile users with various ranges of mobile devices from mobile phones to mobile PC. A mobile peer has a simplified set of functions in order to fit in mobile systems. A mobile peer can join any super peer that is geographically close to him and act as an edge/adhoc peer that allows direct message delivery in the network.

A super peer is a full featured node that operates as a service provider server to serve a set of mobile peers. Multiple super peers can connect to each other in P2P fashion to provide load balancing and scalability. The difference between a super peer and a mobile peer is that a super peer is responsible for creating and monitoring the overlay network and providing services to other super peers and mobile peers. Super peers also provide message propagation and information caching. Mobile peers can switch from one super peer to another whilst maintaining relevant data using query service. The details are provided in Sect. 3.1.

3.1 Components of JHPeer engine

3.1.1 Context service

Context Service handles context-related operations and manages contextual information shared in the peer network. The contextual information includes user profile and dynamic user contextual information such as user location, operation history, CPU usage, calendar and events, etc., which are stored and indexed by caching/indexing component (described in Sect. 3.1.2). Context service is made up of three sub-components: context provider, context manager, and context listener.

Context provider monitors and collects contextual information from various sensors (light, noise, location, temperature, CPU usage, etc.) or external sources such as web services (i.e., weather) and peer contexts.

Context Manager is used for five main purposes. Firstly, analysing context information which is obtained from Context Providers and notifying Context Listener if the contexts change. Secondly, processing context information via associated Context Reasoners. Thirdly, pushing and publishing the collected context information on the peer network. Fourthly, recording and storing the collected context information as persistent data via Cache/Indexing component. Fifthly, handling and responding to context requests or context queries that are submitted by other services and peers.

Context Listeners register with the Contest Manager and respond to context change. A Context Listener can listen for context events from particular peer (user), or listen for specific type of context events from particular sensors.

A Context Representation is defined in JHPeer to store, share, transmit and manipulate various context data in the peer network. There are several ways of modelling context information in a variety of context-aware systems [28] such as key-value models, mark-up schema models, graphical models, object-oriented models, logic-based models and ontology-based models. JHPeer aims to share and reuse context data amongst applications and peers. Simple representation promotes reuse and expandability. The work reported in this paper includes a general expandable Context Representation based on object-oriented (OO) data structure. It benefits from the characteristics of object-oriented technique such as inheritance, polymorphism, abstraction, etc. Developers can implement their own transformation functions in the data object (e.g., toJSON(), toRDF()) to convert context information into required types. Besides, the default implementation of context marshalling is based on XML which facilitates transmission and interoperability.

Figure 2 shows the context representation in JHPeer. The IndexableDoc object is designed to permit Cache Service to index and store documents. It is also a subclass of JXTA Advertisement class that is used for transmitting and publishing context information in the peer network. The subclass of IndexableDoc for handling contextual information includes ContextAdvertisement, DeviceAdvertisement and UserProfileAdvertisement. ContextAdvertisement holds the dynamic contextual information such as location, weather, device/application status, etc. A set of ContextModule in ContextAdvertisement is used to hold contextual information. A ContextModule is an abstract object that is extendable to implement relevant context objects. DeviceAdvertisement is used to hold static device information such as operating system, memory, battery capacity etc. It also includes a set of Hardware modules to hold detailed hardware (sensors) information. UserProfileAdvertisement contains the sensitive user information such as user detail, user calendar, friends, system and application preferences, etc. It holds a set of UserProfileModule that contain user preferences.

In order to demonstrate the application and flexibility of the context representation developed in this research, an ontology model proposed in [37] has been implemented and is illustrated in Fig. 3. It consists of two levels: common context ontology and domain-specific ontologies. The common context ontology contains three fundamental contexts: context, device and user. Designers can develop domain-specific ontologies in different domains such as

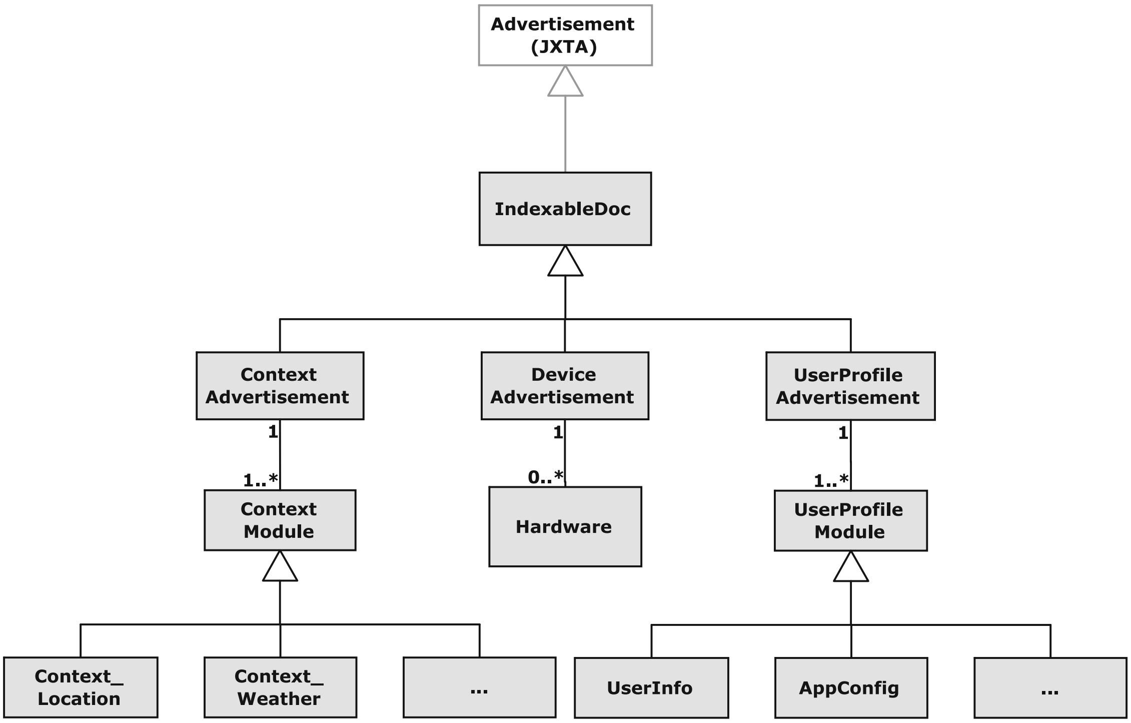
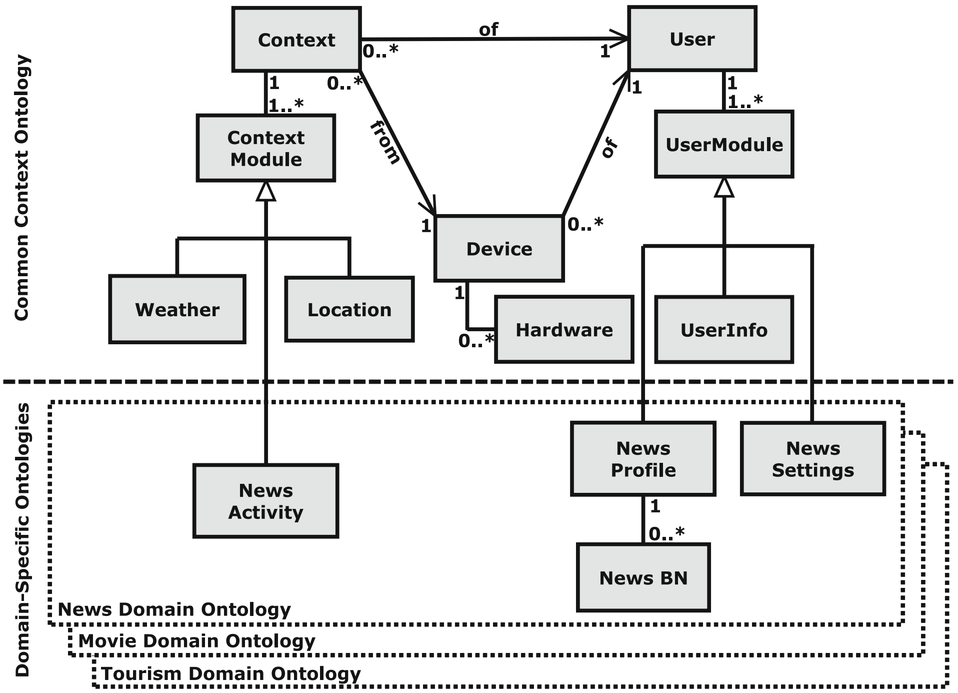


Fig. 2 Context object diagram

Fig. 3 Ontology model using JHPeer context representation



News, Movie, Tourism, etc. For example, in news domain, it may require user reading activity information (e.g., clicked news, reading time, and reading location), news user profile (e.g., news category, average reading time, likely to read at location/time), and news application settings.

3.1.2 Caching/indexing component

Caching/indexing component is responsible for storing/managing/retrieving IndexableDoc object in local storage (e.g., disk, flash drive) and builds index files to enable fast searching. It is implemented on both mobile and super peer with the same set of APIs, and supports different libraries and storage formats. A large volume of stored context information may overwhelm a mobile device; therefore an expiry timestamp is assigned to each IndexableDoc. Caching/indexing component will clean up the expired IndexableDoc from the local storage.

On the other hand, caching/indexing does not have the functionality of data replication. Developers have the flexibility to choose one of the following options: override the JXTA discovery service, implement a distributed cache service, design a new replication service or even override Query Service which is described in the next section.

3.1.3 Query service

Query service is an advanced discovery service that includes four major functions: performing local or remote searching, publishing IndexableDoc document in peer network, propagating query to nearby peers, handling advanced query such as Boolean Query, XQuery, SQL-Liked Query, etc. The reason for implementing Query Service is because JXTA discovery service is only designed for publishing advertisement and performing plain text searching. It can only handle simple keywords query which is insufficient for modern applications.

In normal operation, a mobile peer joins a super peer that is geographically close to it. The Query Service maintains relevant data of mobile peers. When a mobile peer switches from one super peer to another, the new super peer will get user information from the previous one. When a mobile peer is disconnected, updated information will be published to the peer network.

The query service on mobile peers does not provide the functions of propagating queries and collecting results from peers. Therefore, these power/memory consumption tasks are carried out by super peer.

3.1.4 Peer monitor

Peer monitor is designed to check online peer status periodically. This component is essential because mobile peers may frequently disconnect from the peer network in the mobile environment. Since JXTA protocol does not provide a mechanism to keep the real-time online status of peers, a peer status protocol has been developed to retain information about the type of peer, associated services (e.g., news/movie service), peer information (e.g., uptime, ping statistics, device status) and status. The status of a peer is represented in six levels: UNKNOWN, ONLINE, AWAY, BUSY, INVISIBLE, and OFFLINE, which can be assigned by relevant applications automatically or by the user manually. Recommendation services are also based on a user's status. For example, if a user sets his/her status to BUSY, the proactive recommendation services will postpone information pushing to the user. In order to minimize energy consumption, peer monitor in mobile peer will only submit information to super peer when a peer's status changes. Super peers only check mobile peer's status after an inactive period. When the battery energy is low, peer monitor automatically increases the checking time interval.

3.1.5 3rd party services

3rd party services are services that are deployed in the system for providing functionality to users or other system components. For instance, news service enables users to retrieve personalised news based on current users' contexts.

3.2 Application scenario

JHPeer is a generic framework that supports a variety of context-aware applications in mobile environment. Application developers can configure and extend it with application-specific methods to provide context-aware services. The authors have designed and implemented a news recommendation system (PPNews), based on JHPeer framework, to support and assess the feasibility of the proposed approach.

PPNews incorporates news recency, user interest, keywords, peer ratings, user's click history as well as location in a real-time manner. The details are discussed in Sect. 6.

4 Proactive recommendation

4.1 AHP based multi-criteria ranking approach

Recommendation problem is a multi-criteria decision making problem. AHP is a multi-criteria decision making (MCDM) method that has been widely used in various domains [38]. AHP expresses criteria in a hierarchical structure. It derives relative weights of criteria from pairwise comparisons based on the importance of each criterion. There are two common methods in deriving weights: eigenvector method and geometric mean method. However, the eigenvector method suffers from the problem of rank reversal [39]. The developed AHP-MCR approach is based on geometric mean methods. The procedure for ranking recommendations is summarised as follows:

1. Construction of criteria hierarchy, and the alternatives,
2. Construction of criteria comparison pairwise matrix,
3. Derivation of the weights of criteria from comparison pairwise matrix, and
4. Computation of rank for each alternative.

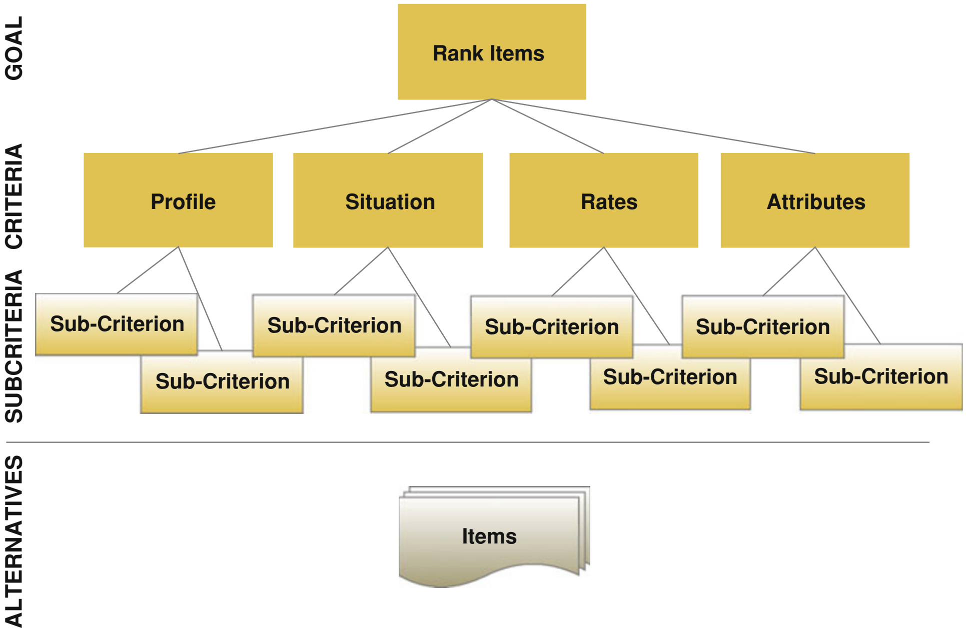


Fig. 4 General recommendation criteria hierarchy

4.2 Construction of criteria hierarchy, and the alternatives

A general criteria hierarchy for mobile recommender has been developed and is shown in Fig. 4. These criteria are selected by investigating the common features of existing researches, such as User Preferences [2,4,15,16,24,40], Contextual attributes [2,4,8,12,16,25,26], community rankings [2,15,16,24] and other attributes [2,4,15,16,24]. Four general criteria are identified and defined: *Profile*, *Situation*, *Rates* and *Attributes*. In addition, nested hierarchy is supported in the construction for flexibility. These four criteria are suitable for different domains such as news, movie, and tourist, etc.

Profile Criterion represents a set of user profile variables used to predict the priority of items which may be of interest to the user. The score of Profile Criterion is computed based on user's preference and user's history. It can also be referred to as R: user(s) × item(s) → ratings in a traditional recommendation system.

Situation Criterion handles various temporal context variables that are relevant to the current environment. Sub-criteria can include emotion, status of user, weather, location, task, with/nearby friends, etc. For example, in a movie domain, a user is going to watch a film with his girlfriend. The recommender should assign a higher score for the films that are suitable for couples.

Rates Criterion is the score based on users' click behaviour and votes. Higher click rates or positive votes mean that an item has a high score. It also refers to an item's popularity.

Attributes Criterion refers to the item's attributes for particular domain or any other relevant attributes. This criterion should be considered as additional factors for recommendation process for specific domain and should not consider any user heuristic information or user preference. For example, in news domain, it may be divided into two sub-criteria: time and site subscription. Time computes the score of items based on recency such that a recent news article has more weight and the weight is reduced as time passes. The score of site subscription criterion is computed based on the number of subscription to a news web site. A higher score is given to the source of news which has more subscription. For movie domain, it can refer to other sub-criteria such as directors, writers, actors and awards. Directors/writers/actors are used to compute the score based on ratings of their movies. A high score is given if their previous movies were highly rated. Awards compute the score based on the number of awards won by the movie, with a high score being given to a movie that has won an award.

4.2.1 Applying AHP-MCR in different domains

In order to show the flexibility of the AHP hierarchy model developed in this research, additional attributes that could be used to design criteria for the different domains are presented in the Table 1 below.

4.3 Construction of criteria comparison pairwise matrix

In the literature, the comparison pairwise matrix (CPM) (p_{ij}) for each criterion pair i and $j$is given by $p_{ij} = C_i/C_j$, where C_i and C_j are the user assigned numerical judgment (value: 1–9) of criterion i and j, respectively [41].

However, the importance of the criteria which can only be assigned by the user is not flexible enough for a

Table 1 Example of criteria in different domains

Type	Attributes
In movie domain, the possible attributes for recommendation process can include	
Profile	Address, age, occupation, preferred cinema/location/genre
Situation	Watch location, with whom, watch time, mood
Rates	Ratings/votes
Movie attributes	Movie name, director, writers, actors, release date, country, awards, gross revenue, medium (TV/DVD/Blu-ray), cinema, genre
In tourist domain, the possible attributes can include	
Profile	Address, age, occupation, social economic grade, personality, preferred activity types
Situation	With whom, mood, location/distance, visit time (opening time), transport, weather
Rates	Ratings/votes
Tourist attributes	Cost, suitable age ranges
In news domain, the possible attributes can include	
Profile	Address, age, occupation, social economic grade, interested news categories, interested topics, preferred newspapers/websites
Situation	Location, weather
Rates	Ratings
News attributes	Publish time, length of article

recommendation system. By adopting AHP-group decision making [42], the assignment of criteria pair importance can be generally classified into three different categories: individual-based, group-based and mixed approach.

Individual-based approach assigns the importance of criteria by user assignment or computes it based on user's history. The user assignment can be done by implementing a scale bar or from a questionnaire from a user interface. In contrast to an explicit method, it can be computed using a user's history. For instance, each recommendation that has been read by a user should contain the score for each criterion. The numerical value of criterion i (C_i) is the total number of the highest value (n_i) of criterion amongst all criteria (C_n) (in same hierarchy level, where n is the sub-set of all Criteria N) in a user's history which is represented as: count $(\max_{n\in N} n_i)$.

Group-based approach adopts the techniques in AHP-group decision making called aggregation of individual judgments (AIJ) [42]. The importance of the criteria is aggregated from users. There are three possible methods:

Method 1: By aggregating all users' numerical judgment (assigned or/and computed judgment) computed for each criteria.

Method 2: By aggregating numerical judgment (assigned or/and computed judgment) from users that have similar contexts. It can apply common similarity methods suchas Pearson correlation coefficient, cosine similarity, Euclidean distance, k-nearest neighbours, clustering (e.g., k-means clustering), etc. to find users.

Method 3: By applying machine learning techniques to calculate likelihood values such as Bayesian prioritisation procedure [43] and fuzzy preference [44,45] on incomplete data sets.

Mixed approach enables the combination of both individual-based and group-based approaches allowing assigned personal importance and global users' importance to be taken into account. The final importance is computed by arithmetic means. For example: a set of numerical value of criteria i is defined as X_n, where $n = \{1, 2, 3\}$, $n = 1$ is user assigned individual numerical value, $n = 2$ is normalised numerical value which is computed from user history which is defined as count $(\max_{n\in N} n_i/\text{total})$ 10, $n = 3$ is the group numerical value between 0 and 10. The importance of criteria i is the arithmetic mean of X_n that is represented as $C_i = \overline{X_n}$.

4.4 Derivation of the weights of criteria from comparison pairwise matrix

Once the comparisons matrices are specified, the weight of criteria (w_i) is computed using the geometric mean as follows:

$$w_i = \left(\prod_{j=1}^{n} p_{ij}\right)^{1/n} \tag{1}$$

where n is the number of criteria in the same hierarchy level and p_{ij} is the numerical value from previous step.

4.5 Computation of priority for each alternative

The priority of each alternative requires normalisation before the computation of the final score. The alternative (item) normalised priority (N_{ij}) is calculated as follows:

$$N_{ij} = r_{ij} / \max_k r_k \tag{2}$$

where r_{ij} is the assigned priority of criteria (p_{ij}) by any rating-based algorithm and is divided by the maximum value of the priority (r_k) in the corresponding criteria (p_{ij}).

The last procedure is to synthesise all scores across all criteria in each hierarchy-level from the bottom to the top to determine the final priority. The score of an item ($score_i$) is computed as follows:

$$\text{score}_i = \sum_{j=1}^{n} N_{ij} w_j \tag{3}$$

where N_{ij} is the normalised score of the criteria from equation (2) and w_j is the weight of criteria from equation (1).

Only top-k alternatives from the ranked list will be pushed to the users' mobile devices based on implemented recommendation pushing approach which is explained in the next section.

4.6 Pushing of recommendations

For pushing recommendations, four types of approach are identified. They are preference-based approach [46], knowledge-based approach [4,5], content-based approach [47] and hybrid approach.

Preference-based approach uses user assigned preference to push recommendations. For instance, a user may wish to receive recommendations in a particular time period or to receive recommendations as soon as possible. Furthermore, relative preference of a user (new user) can be predicted by collaborative approach from peer users.

Knowledge-based approach uses various types of knowledge to predict user needs or suitable moment to push recommendations. The knowledge could be a set of logical rules (association rules/decision tree) assigned by experts or a set of previously stored data (cases/experiences) using machine learning techniques such as Bayesian network, fuzzy logic, etc.

Content-based approach pushes recommendations to a user based on user's history. Similar recommendations that a user preferred (pushed and read) in the past will be pushed. However, this approach may not be suitable in some domains because sometime a user may not want items that are too similar. For example, in news domain, a user has just read news about "new phone release"; he may not want to be notified if there is similar news about that "phone" again.

Furthermore, recommenders can combine the approaches to form a hybrid approach that can potentially provide a better performance in pushing relevant information at the right time.

5 Using Bayesian network to eliminate cold-start problem

Typical recommenders have to face the data sparsity problem for new users which causes difficulty in predicting users' interests and computing criteria score (e.g., Profile). Traditionally, there are two methods that are used to calculate a user's interests. One is based on the user's history, which is a content-based method, and the other is based on group history, which is a collaborative filtering method. However, the content-based approach requires analysing a large amount of user's interests and may suffer from the problem of pushing very similar recommendations to the user. Although the collaborative filtering method can offer diverse recommendations, a simple collaborative based recommender may not meet the specific user's interests.

A Bayesian network is a probabilistic graphical model that combines the advantage of CF and CBF. It means for a new user, the system will use group profile data and generate a global BN that not only can solve the data sparsity problem, but also can provide a variety of options. After the user uses the application for a while, the system will revert to using the contents of the user's profile to make recommendations that are more suitable for that particular user. Additionally, BN could provide real-time prediction to achieve the objective of just-in-time personalisation because BN requires a smaller memory and provides faster computation than other CF/CBF techniques [48].

Bayesian network is used to represent and predict an uncertain domain in a graphical structure [49]. It is a directed acyclic graph. The nodes represent the conditional independence relationships amongst the variables. The arcs represent dependent variables to their parents. The chain rule of probability theory that defines the joint probabilities of n nodes in a BN is calculated as follows:

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\text{Parents}(X_i)) \tag{4}$$

In the recommendation system, BN can be used to estimate a user's interest and predict an uncertain user's situation based on the user's or group's previous activities and history. To accomplish this objective, three phases are designed to fulfil the requirement of a BN which are construction, prediction and revision phase.

5.1 Construction of Bayesian network

The Bayesian network construction is based on Pearl's Network Construction Algorithm [49] that follows three steps: selecting relevant variables, specifying the Bayesian network and defining the conditional probability table.

We classified four types of relevant variable which can be applied to a recommendation domain: demographic, situational, psychographic and domain specific variables.

The demographic variable contains gender, age range, occupation status, etc. These attributes can influence the user's interests to a large extent. For instance, in news domain, men are more interested in sport news than women; in movie domain, women prefer romantic movies than men. The situational variable includes location, usage period, time spend, weather, with whom etc. For instance, in movie domain, if a user is dating, the chance of choosing a romantic movie is greater than choosing an action movie. The psychographic variable is considered as personality, lifestyle, characteristics, values, attitudes, hobbies, and general interests. Personality, lifestyle and characteristics can determine how a person treat or react to a situation. The domain specific variable is comprised of different attributes in each domain. For example, in movie domain, a developer needs to consider movie genre, actor etc, whilst in tourism domain, the possible attributes are tour motivation, cost and activity types.

After the relevant variables are selected, there are two approaches that can be used to specify the BN structure. One is to join variables manually. The other is to learn the BN structure automatically via historical data. The typical examples are Cooper & Herskovits' K2 algorithm and Minimum Description Length (MDL) [49].

Finally, the conditional probability table (CPT) of each variable has to be defined. There are several techniques available to developers. Firstly, the CPT can be assigned manually. Secondly, it can be carried out by a questionnaire survey. Thirdly, it can be built using sampling algorithms based on collected data.

5.2 Prediction of user interest

The major task of this phase is to compute the posterior probability distribution of a target node using any given evidence nodes. The computation is based on a chain rule of probability theory given by Eq. (4). A specific posterior probability value is used to estimate the user's interest in AHP-MCR approach. Soft evidence is also employed to improve the prediction instead of using a user's local BN. An example is shown in Sect. 6.8.2.

5.3 Revision of BN (Offline Process)

Revision of BN is divided into two parts: revision of global BN and revision of local BN. When a user is new to the system, global BN is applied. Whilst the user's history is accumulated, the user's local BN is being generated. Both BNs are revised based on incremental learning [49]. Besides, incomplete data can be resolved by Gibbs sampling or expectation maximization (EM) algorithm [49]. Therefore, the system can effectively eliminate the problem of data sparsity for new users.

In addition, the revision process is carried out as an offline process that may be scheduled on an hourly or daily basis, since updating the BNs may cost a lot of computation power.

6 Implementation of a news recommender

In order to assess the feasibility of the JHPeer framework and the AHP-MCR approach developed in this research, a Proactive Personalised News (PPNews) application was developed. Since it is based on pure Java, it can ideally run on any Java-enabled mobile devices, including Android mobile platform.

PPNews is designed to provide just-in-time personalised news services to meet mobile users' needs [50]. The key features include: proactive, context-aware and just-in-time personalised news recommendations. It is built on top of JHPeer framework that incorporates user context and news article attributes for automatically pushing real-time news which are considered relevant to mobile users. The major benefit of using JHPeer is to allow mobile devices listening for "push" news items. The delivery of news items are faster than using traditional push-pull methods. Moreover, a large number of servers can be used in a hybrid P2P network, which provides the capability of load balancing and supports distributed services to deal with a large number of users and to create collaborative communities.

6.1 System overview

Figure 5 illustrates the information flow between a mobile device and the JHPeer server. A mobile device connects to one of the servers and submits a user's profile. The context service in the mobile device will automatically push context information. The servers constantly collect real-time news articles from news contents providers via really simple syndication (RSS). Instead of asking users to input a query directly from their mobile device, the server filters and ranks new news articles based on a user's profile, location, usage patterns, peer ratings and news article attributes. A user can change the weights of each ranking component depending

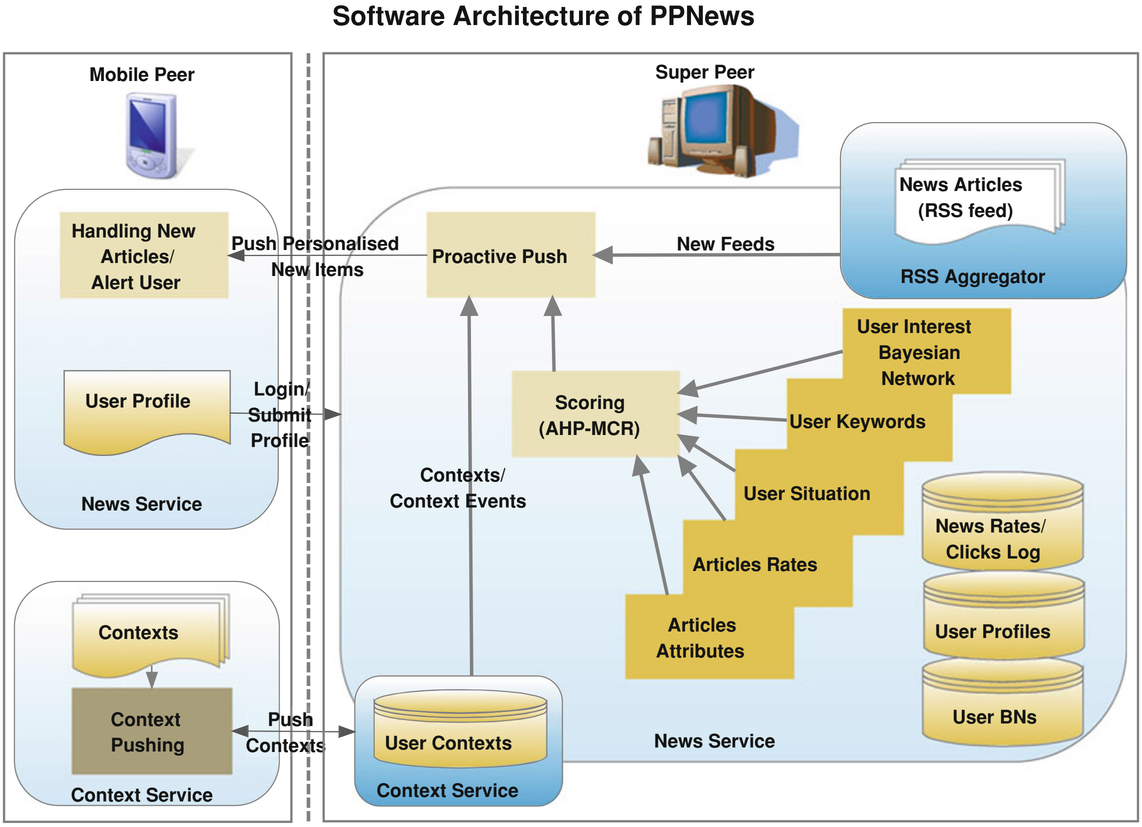


Fig. 5 Information flow between mobile device and JHPeer server

on personal interest. Top-k (the value of k can be changed by user) news stories of each category will then be proactively delivered to the user's mobile device. During the interactions with the user, the system collects the user's reading patterns, location, peer activities and use these contextual information to predict the user's interests and update the user's profile.

6.2 Context service for PPNews

Different types of service may require different levels of contextual information. Therefore, the context service may have varied implementation between applications. JHPeer provides a basic implementation of context services, context provider and context listener. Although both super peer and mobile peer share the same set of APIs, the actual implementations are different. A mobile peer mainly captures low-level contexts (sensor data), while super peer is responsible for processing the low-level to high-level contexts. Through this implementation, a mobile device maintains a longer battery life, since complex computation is handled by super peer.

6.2.1 Mobile peer

The context service in mobile peer is implemented to push low-level contexts to super peer via context advertisement periodically. The implementation of context service supports filtering and provides (particular) context events for other services through Context Listeners, where the context events are generated by context providers or super peer contexts push. The context providers in PPNews capture the context data from GPS/accelerometer sensors, CellID (location) and device's calendar (current meeting or not).

6.2.2 Super peer

The context service in super peer is implemented to receive and aggregate contexts from mobile peers. In current implementation, simple context reasoning is implemented to find relevant user location (address) and weather information. Since high-level contexts such as a "user is currently at home/work" is not useful in news domain. The notification to context listener is the same as the one in the mobile peer. The context provider is designed to receive contexts pushed from mobile peers in PPNews.

6.3 Caching/indexing service

The caching/indexing service is implemented using different libraries in PPNews. In super peer, Apache Lucene [51] library was implemented to provide fast indexing and search

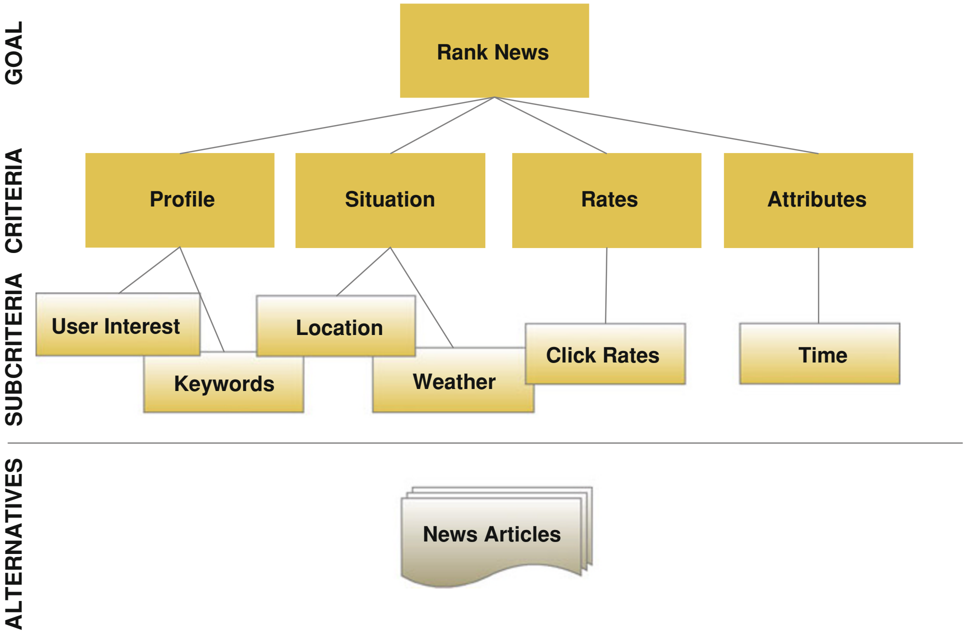


Fig. 6 PPNews AHP hierarchy

facilities. News items can be retrieved from the local storage quickly with the additional information like term frequency (TF) and inverse document frequency (IDF) which benefit the computation of recommendations. In mobile peer, Perst [52] object-oriented library was used to store any IndexableDoc object (e.g., UserProfile, News Articles/Channels) for fast retrieval. It includes a basic full text index for fast searching. In addition, Perst library can run on any Java-enabled phones.

6.4 Other services on JHPeer

The query service is implemented in PPNews and supports Boolean query.

Peer monitor is implemented to check a peer's online status periodically.

6.5 Proactive news push

PPNews is mainly based on preference-based approach. Users can configure the preferred news categories/sites to receive news periodically/instantly or as determined by the system. The system determines the user's preference by computing a threshold (likelihood) value by analysing a user's history. If the score of new article(s) is equal to or greater than the value, then the news article(s) will be delivered to the user.

6.6 AHP criteria hierarchy in PPNews

The criteria selected for PPNews is shown in Fig. 6. These criteria are selected based on the common ranking factors in news domain. It is also suitable for a new news recommender that does not require large amount of historical data. The detail of each criterion is described in the next section.

The criteria comparison pairwise matrix is implemented based on individual-based approach that combines both explicit (user preference) and implicit (automatic learning through user activity) methods to enable automatic priority adjustment. Only the weight of the four major criteria has been made available for users to adjust on the GUI because a large number of questions may discourage some users. The importance of sub-criteria is assigned a default value (i.e., 31) which refers to equal importance.

6.7 Computation of profile criterion

Profile criterion is the key factor for news ranking due to frequent updating of news web sites. To rank new articles, the long-term and short-term user histories should be analysed. In PPNews, the long-term factor is the particular news category that a user is interested in. For the short-term factors, news keywords are used to rank the news stories. Only the keywords of recently read news are used as a user's interest may change over time [53].

The score of a profile criterion is computed by combining the scores of all sub-criteria and their weights. The sub-criteria include user interest and keywords.

6.8 Computation of user interest criterion

In PPNews, a Bayesian network (BN) is developed to predict the level of interesting news categories for a particular user and also resolve the problem of cold-start for new users. All the CF/CBF techniques require analysing a large number of users' histories. We construct the BN via a summarisation of empirical data from recently published researches [54,55] in news domain.

6.8.1 Construction of BN

The selection of relevant variables for PPNews is based on theoretical and empirical study in news domain. By reviewing a number of news web sites and published research works [15,54–59], five variables were identified as follows:

Gender: The preferred news is varied because of a user's gender [54,55,57]. For instance, a man is more interested in sport than a woman. Obviously, two states are used for this variable: Male and Female.

Age range: It is a common demographic factor that affects the choice of news category. Different age groups have different preferred news category [54,55,57]. For this variable, common age ranges are used: 0–24, 25–34, 35–44, 45–54, 55–64, 65+.

Occupation status: Working status has an effect on a user's interests [54,55]. For this variable, four basic occupation states are considered: full-time, part-time, unemployed, and retired.

Social economic grade: The social economic grade is influential in the category of news users preferred [54,55]. For this variable, four states are considered: AB (£35-50k+), C1 (£25-35k), C2 (£15-25k), DE (£0-15k).

Category (news): There are many ways to organise news articles. For simplicity, this research only considers the news at the top level rather than organising them into different levels of hierarchy. The category is organised in common states [15,55,58,59]: Headlines, business, politics, entertainment, health, science, technology, sports, world, UK (nation).

The global BN is developed and the relationships between the variables are established as shown in Fig. 7. The values in the CPT for each variable are assigned based on the summarisation of empirical data from [54] and [55]. Four variables: gender, occupation status, age range and social economic grade are root nodes, which influence the category node. If the probability distribution of any of the three variables is changed, the probability distribution of the category node also changes.

6.8.2 Predicting user interested category

The posterior probability distribution of the category node is calculated using given values from users' demographic contexts. A user's news interests category is predicted based on the user's personal information (gender, occupation status, age, and social economic grade). Figure 8 shows the predicted result of posterior probability distribution for category node given the user's profile (*gender = Male, age range = 0–24, occupation = part time, social economic grade = C1*).

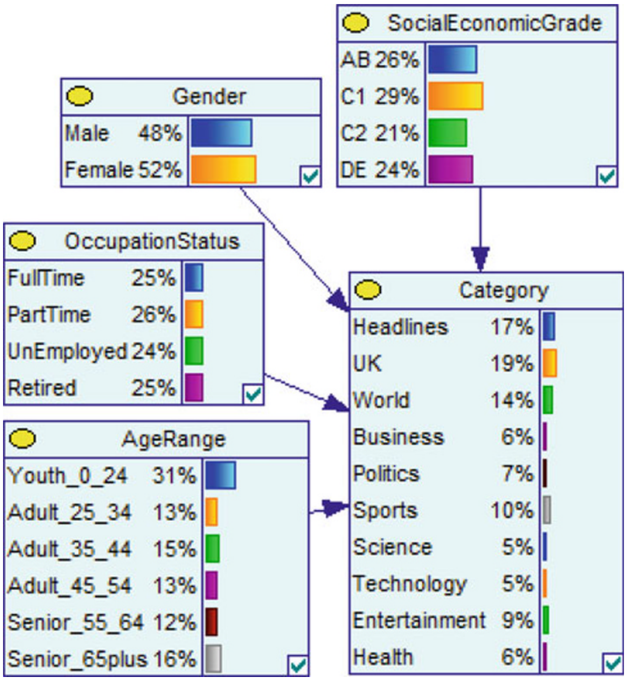


Fig. 7 Prior probability distributions of News BN

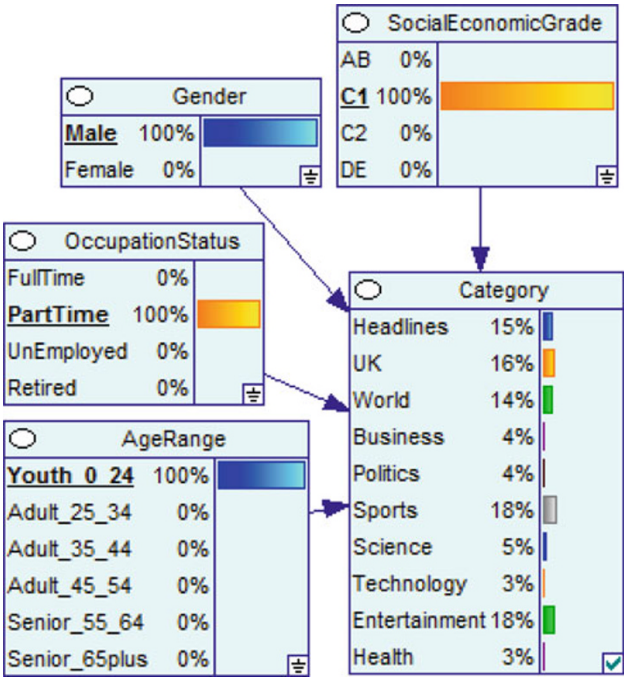
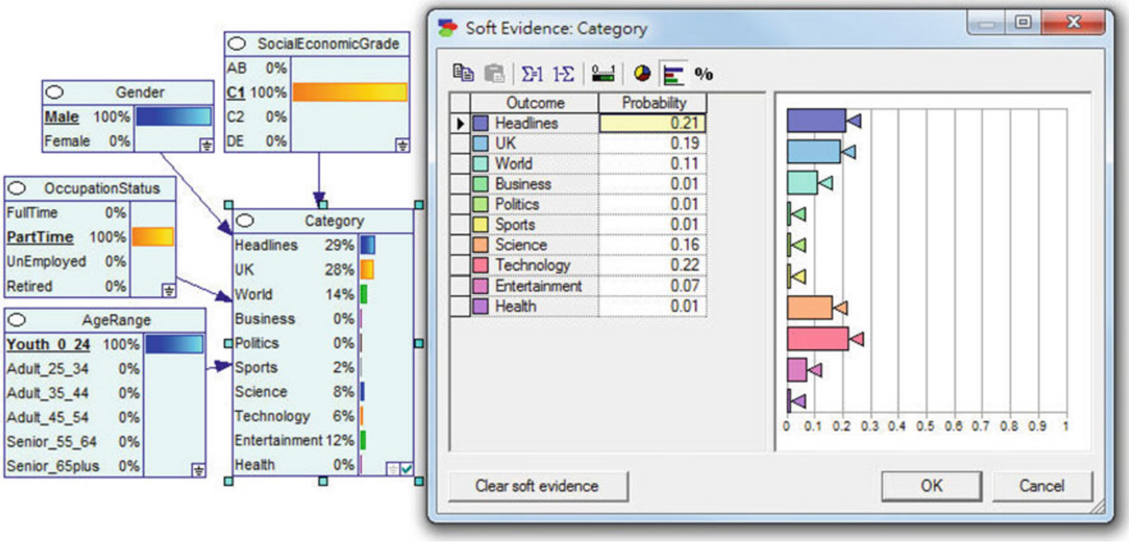


Fig. 8 Posterior probability distributions based on user demographic contexts

On the other hand, if the user is not willing to provide any demographic contexts for privacy reason, the prior probability distributions of News BN will be used.

When a user's past reading history is accumulated, the soft evidence method is used to predict the user's news interest. Figure 9 shows an example of applying the soft evidence.



SocialEconomicGrade	
AB	0%
C1	100%
C2	0%
DE	0%

Gender	
Male	100%
Female	0%

OccupationStatus	
FullTime	0%
PartTime	100%
UnEmployed	0%
Retired	0%

AgeRange	
Youth_0_24	100%
Adult_25_34	0%
Adult_35_44	0%
Adult_45_54	0%
Senior_55_64	0%
Senior_65plus	0%

Category	
Headlines	29%
UK	28%
World	14%
Business	0%
Politics	0%
Sports	2%
Science	8%
Technology	6%
Entertainment	12%
Health	0%

Soft Evidence: Category

Outcome	Probability
Headlines	0.21
UK	0.19
World	0.11
Business	0.01
Politics	0.01
Sports	0.01
Science	0.16
Technology	0.22
Entertainment	0.07
Health	0.01

Fig. 9 Final posterior probability distributions with soft evidence

Based on a user's past click behaviour, the probability of category (C) node with soft evidence (C') is calculated as follows:

P(C = Headlines) = 0.15, and given 21% of soft evidence for Headlines variable. Using Bayes' Theorem to perform the inference, it is equal to:

$$\begin{aligned}\mathrm{Bel}(C' = \text{Headlines}) &= \alpha P(C' = \text{Headlines}|C \\ &= \text{Headlines})\, P(C = \text{Headlines}) \\ &= \alpha 0.21 \times 0.15\end{aligned}$$

Since $\mathrm{Bel}(C' = \text{Headlines}) + \mathrm{Bel}(C' = UK) + \mathrm{Bel}(C' = \text{World}) + \cdots + \mathrm{Bel}(C' = \text{Health}) = 1$, this gives us $\alpha = 9.3213$. Therefore, $\mathrm{Bel}(C'\text{Headlines}) = 9.3213 \times 0.21 \times 0.15 = 0.2936$, as shown in Fig. 9.

By applying the soft evidence in the BN, personalised posterior probability distributions of Category node can be generated. This posterior probability distribution will be used to define the priority of User Interest criterion (r_{ij}).

6.8.3 Revision of BN (offline process)

In PPNews, the structure of BN is fixed and the data is complete. The news BNs are revised based on parameter learning [49] in super peers.

6.9 Computation of keywords criterion

Keywords are a list of words that are used as a short-term factor for news ranking. The keywords list is generated based on users' activities in past 7 days. In PPNews, keyword-based searching techniques are designed to compute the score of news stories. Based on users' past activities, all words in the clicked news title will be indexed. Only top-k keywords are used in the scoring process. The scoring process is computed using term frequency-inverse document frequency (TF-IDF) which is shown as follows:

$$\begin{aligned}w_{m,n} &= tf_{m,n} idf_m,\ tf_{m,n} = n_{m,n} / \sum\nolimits_k n_{k,n},\ idf_m \\ &= \log N / n_m\end{aligned} \qquad (5)$$

where $n_{m,n}$ is the number of occurrences of the considered term, t_m, in news story (n), and the denominator is the sum of number of occurrences of all terms in news stories. *idf* takes the logarithm of the number of all documents, N, over the number of documents containing the term n_m. The priority of article ($r_{i,j}$) is equal to the sum ($\sum_m w_{m,n}$) of the keywords' weights ($w_{m,n}$).

To provide optimised results to new users, global keywords are applied.

6.10 Computation of situation criterion

Situation criterion currently includes location criterion and weather criterion. Using Geocoding service in super peer, it converts latitude and longitude which is obtained from a device's GPS to textual location (address) for carrying out term weighting process on the news articles. For example, if a user is located in Portsmouth, Hampshire, UK, the system will go through similar process as Keywords Criterion using TF-IDF. The terms used in this example will be Portsmouth, Hampshire, and UK. If a news article contains these terms, it should be given a higher score. Moreover, other situation contexts may be relevant in news domain such as weather contexts. For instance, Weather RSS feed can be used to retrieve textual weather information (e.g., flooding) of the user's location. The TF-IDF value of each keyword in weather information is calculated. If a news story contains the weather keywords (e.g., flooding), it will be given a higher priority.

6.11 Computation of rates criterion

Rates criterion is used to determine the importance of a news story. Currently, click-rate of news items is implemented. It is based on simple counting mechanism. Higher scores are applied to most clicked items. Votes of news will be considered in future to complement click-rate.

6.12 Computation of attributes criterion

Attributes criterion refers to time recency since only predefined RSS websites are currently allowed. The time recency of news article is computed as follows (adopted from [15]):

$$r_{i,j} = -(t_{\text{present}} - t_{\text{posted}}/t_\beta)^2 + 1 \qquad (6)$$

where t_{present} is the current timestamp, t_{posted} is when the article was posted and t_β is the base timestamp which is equal to 24 h. For instance, if an article is posted at 13:00 GMT and the current time is 15:00 GMT, the weight is equal to $-((15-13)/24)^2$ +1 = 0.9931. The score will be zero if the posted time is greater than the base time (t_β).

7 Experimental analysis

The goals of the experimental analysis were twofold: first, to evaluate the performance of JHPeer framework on mobile devices and; second, to evaluate the performance of the proposed AHP-MCR approach.

7.1 JHPeer framework

The analysis of the JHPeer framework mainly focuses on mobile peers, since the feasibility and usefulness of JHPeer framework on mobile devices is more important than super peer. Super peers are mainly deployed on desktop/server computer which have extensive resources including powerful processing and most likely persistent broadband connectivity. The performance of a super peer is related to the design of the services such as algorithms implemented and control of load balancing. Besides, some researches that have already been carried out to examine the scalability and performance of JXTA on desktop and mobile environments are reported in [60] and [61] respectively. Therefore, this paper primarily analyses the performance on mobile information delivery to show the usefulness of a proactive recommendation application.

The evaluation was carried out using a HTC Desire Smartphone (Android 2.2, 1GHz CPU, 576MB RAM, HSPA/WCDMA/GSM, SLCD Screen Brightness LOW, Network Location Service) and the application size of PPNews is 1624KB (without obfuscation). The Android platform was selected since it can retrieve detailed battery status easily and it is a relatively powerful mobile computing platform. On the other hand, the Super Peer is installed on a Laptop with Intel Core 2 T7200 CPU (2.00 GHz, 4 MB Cache, 667 MHz FSB) and 3 GB of memory that runs on Windows 7 SP1 (64-bit) and Java 6.

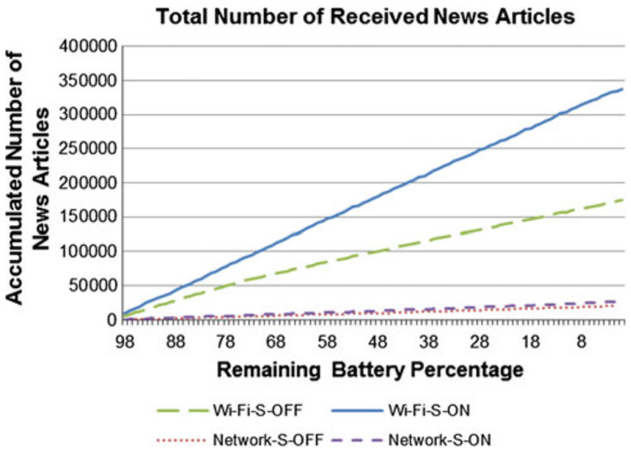


Fig. 10 Number of received news articles versus battery

In this experiment, PPNews was configured to push a random news article to the mobile device continuously in order to evaluate the performance of JHPeer platform in different settings. The settings include: Wi-Fi with Screen OFF, Wi-Fi with Screen ON, Cellular Network with Screen ON, and Cellular Network with Screen OFF. The Cellular Network used was O2 UK and tested in indoors. It was necessary to test the performance for both Screen ON and OFF, since the new generation of smartphones will manage and lower the power consumption when the screen is turned off. The second reason is because JHPeer framework will also pause or stop unessential modules/services (i.e., Context Providers) to preserve battery.

Figure 10 shows the accumulated number of received news articles before the battery ran out. The results show that performance on Wi-Fi networks with Screen ON is the best (Wi-Fi-S-ON 357172 articles, Wi-Fi-S-OFF 184658 articles, Network-S-ON 28782 articles, Network-S-OFF 21945 articles). The reason is because the Android system reduces the CPU frequency when the screen is switched off to save battery power and also slows down the processor speed.

Figure 11 shows the details of KB received for each battery percentage. Since the articles were pushed randomly, the lengths of the articles were not the same. The average KB received for Wi-Fi-S-ON was 6847KB, Wi-Fi-S-OFF was 3574KB, Network-S-ON was 613KB, and Network-S-OFF was 470KB.

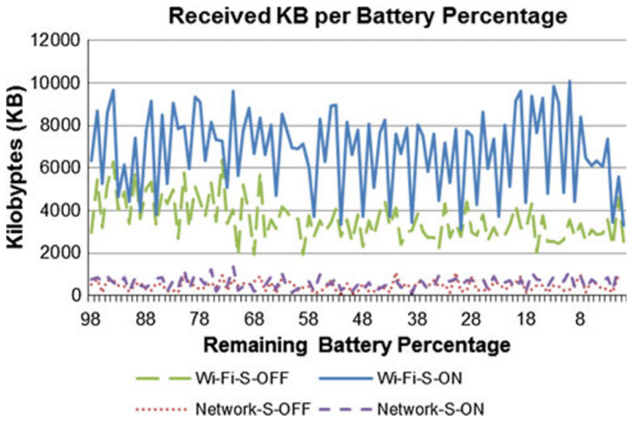


Fig. 11 Received KB per battery percentage

Table 2 Average time used in system operation

	Average message handling time (ms)	Average sending time of response message (ms)	Average operation completed time (ms)
Wi-Fi-S-ON	9.89	2.58	50.71
Wi-Fi-S-OFF	29.35	3.5	370.23
Network-S-ON	22.48	4.31	642.25
Network-S-OFF	22.03	4.12	641.32

Table 2 shows the average time (in millisecond) to deliver news to the mobile device. The average message handling time represents the time used to process the news article from XML string to news item object. The average sending time of a response message represents the time that the smartphone used to send a reply message to a server for notifying receipt and logging record. From the results, it shows that the average time to deliver a news item to mobile device was approximately 640 ms, which is acceptable for a recommendation system.

This experiment shows that JHPeer framework is suitable for developing context-aware information applications. The battery life is up to 5 h, even it is receiving messages continuously. In a standby mode, the battery could last two days or more.

7.2 AHP-MCR Performance

We evaluated the user experience and effectiveness of the AHP-MCR approach in terms of the accuracy of news rating. We compared AHP-MCR with the traditional recommendation techniques.

As no public dataset on context-aware recommendation is available, we created a dataset for our comparative study. To increase the number of participants, we implemented PPNews in J2ME CLDC, Android and Blackberry platforms. 23 users participated in this experiment. 24 RSS News web sites were defined to allow user subscription. A feedback mechanism in PPNews application has been implemented. When the users run the PPNews application on their mobile phones (Android and Blackberry phone are used), news articles were automatically pushed to the users depending on their individual contexts and settings. Users' activities such as news clicking are automatically collected by Context Provider component and processed on super peer. Participants gave ratings for news story on a 1–10 scale, which included overall, profile, situation, rates, and attributes. A result of 352 feedbacks was collected from participants.

Table 3 Experimental result of MAE

	MAE
Standard user-based kNN CF @ all users	0.9675
Standard user-based kNN CF @ 3 neighbourhood	1.1261
Simple TF-IDF based CBF	0.9054
AHP-MCR	0.8504
Profile: profile' rating	0.8942
Situation: situation' rating	0.9159
Rates: rates' rating	0.9045
Attributes: attributes' rating	0.6908

Standard user-based CF approach which uses cosine-based k-nearest neighbour method, and simple CBF approach which uses TF-IDF method were employed as baselines to compare the performance of AHP-MCR approach with other recommendation approaches [12]. In order to obtain evaluation results for user-based CF approach, random data was selected for prediction purpose. In the literature, there are numerous evaluation metrics for measuring recommender performance [62] such as Mean Absolute Error (MAE), Precision, F-measure, ROC. In this paper, we focus on using popular evaluation metrics: MAE and Precision@Top-N.

Mean absolute error method is used to measure the prediction performance of context-aware multi-criteria recommenders by analysing the difference error between the predicted score and the score which is given by user. The MAE is computed by summing all absolute errors between the predicted ratings and user ratings. Then it is divided by the total number of ratings. The lower the value, the better the result is. Equation (7) shows how MAE is calculated.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\mathrm{predict}_i - \mathrm{user}_i\right| \tag{7}$$

Table 3 summarises the MAE results of several recommendation approaches on PPNews dataset collected from the participants. The result shows that AHP-MCR approach performs slightly better than standard user-based CF and CBF techniques. Simple CBF approach even performs slightly better than CF approach in news domain. In addition, we compared the predicted ratings of 4 major criteria with user feedback ratings (on items which had these values). The MAE of Profile, Situation, and Rates have similar performance. The MAE of Attributes is relatively good especially as time recency is easy for users to rate.

Precision@Top-N is used to measure the accuracy of the ranking algorithms in predicting top-N items which user would rate highly. The Precision@Top-N is computed using the number of highly rated recommendations over the top-N number of recommendations as shown in Eq. (8). The highly rated recommendations in this study are taken as any item which is rated with a score of 5 or more by the user.

Table 4 Experimental results of Precision@Top-N

	Precision@Top-5	Precision@Top-10
Standard User-based CF @ all users	0.6511	0.6186
Standard User-based CF @ 3 neighbourhood	0.5845	0.5508
Simple TF-IDF based CBF	0.6042	0.5151
AHP-MCR	0.6993	0.6287
Profile	0.5915	0.5315
Situation	0.2942	0.2348
Rates	0.3451	0.3007
Attributes	0.4629	0.3956

$$\text{Precision@Top-}N = \frac{\text{Number of highly rated recommendations}}{\text{Total number of recommendations}} \tag{8}$$

Table 4 summaries the results of Precision@Top-5 and Precision@Top-10. In this experiment, the precision of each major criterion is calculated to show the effect of contextual criteria on the AHP-MCR ranking process. The precision is obtained using the average precisions of evaluated users. The AHP-MCR approach has better precision than standard user-based CF and TF-IDF based CBF. The standalone criterion performs relatively poor especially for Situation and Rates criteria. This is because not all news articles contain the situation information and not all news articles which a user may be interested in are highly clicked by other users.

In conclusion, this study shows that using the AHP-MCR approach to embed context-aware computing can increase the quality of a recommendation system (Fig. 12).

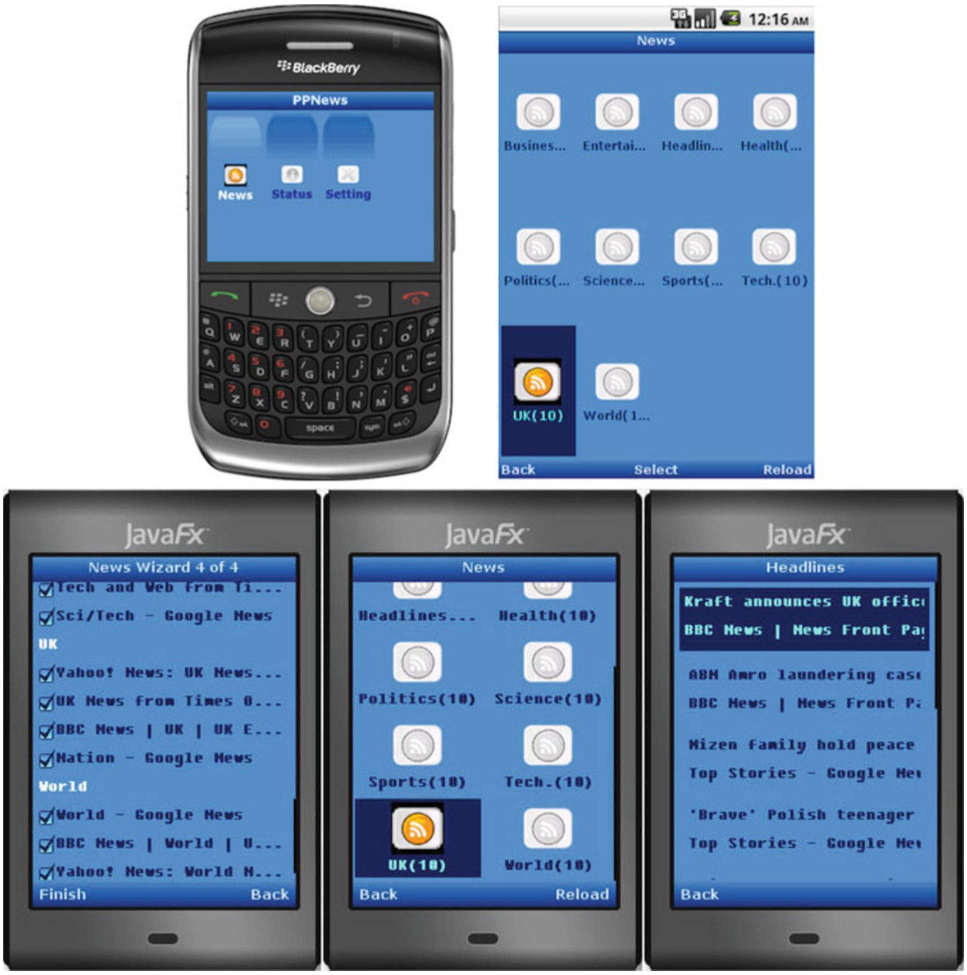


Fig. 12 PPNews screenshots on BlackBerry, Android and standard Java emulators

8 Conclusions

This paper has presented the JHPeer context-aware framework that supports a variety of context-aware applications in mobile environment. Since JHPeer framework is based on P2P technologies, the system developed and described in this paper can be scaled to handle more users by adding super peers. The paper has also presented a general AHP Multi-Criteria Ranking approach to rate recommendations in various domains that supports embedding any rating-based algorithm such as collaborative filtering and content-based filtering for estimating the final scores. The weights of the various contexts (criteria) are automatically adjusted using individual-based and/or group-based (group decision making) assignment. Additionally, a Bayesian Network algorithm has been applied to solve the cold-start problem inherent in recommendation systems. A strategy of how to use the Bayesian network in recommendation services to predict user interest and combine the CF/CBF techniques which make use of user preference and community history has been provided. The paper has discussed the design and implementation of a Proactive Personalised News recommender, PPNews, which is based on JHPeer framework and AHP-MCR approach. Unlike the traditional recommendation systems that usually requires users to submit query explicitly, the PPNews automatically delivers news articles to mobile user based on the user's interests and news contents.

The evaluation results show improvement in recommendation quality. However, it also shows that there has been only a relatively small improvement in recommendation accuracy. This relatively small improvement in the accuracy of the recommendations could be augmented by applying more complex algorithms in the AHP-MCR such as singular value decomposition (SVD) and PLSA algorithms and, using large datasets to train the system. In addition, the system can use a suit of algorithms that are individually optimised for each domain to achieve high recommendation accuracy. Any of this extension would be supported by the generic hybrid P2P framework, JHPeer, which has been developed as part of the research reported in this paper.

As the future work, we will deal with following issues:

Firstly, the current JHPeer framework is designed for delivering information in the form of message. However, the response time is not suitable for transmitting time-sensitive content such as voice and video. A simple and fast communication API for audio and video data is needed.

Secondly, we will explore simple mechanisms which can provide secure communication. A possible solution is to implement a membership service to enable user/service registration and authentication. We will study how to protect user context privacy at different levels.

Thirdly, the JHPeer framework supports push recommendations. A user information needs module is desirable to detect user information needs and request recommendation services in application layer to push just-in-time recommendations to users. Bringing this module may simplify applications development using the JHPeer framework.

References

1. Wang C-Y, Wu Y-H, Chou S-C (2010) Toward a ubiquitous personalized daily-life activity recommendation service with contextual information: a services science perspective. Inf Syst E-Bus Manag 8:13–32
2. Woerndl W, Brocco M, Eigner R (2009) Context-aware recommender systems in mobile scenarios. Int J Inf Technol Web Eng 4:67–85
3. Abbar S, Bouzeghoub M, Lopez S (2009) Context-aware recommender systems: a service oriented approach. In: Presented at the VLDB 2009. Lyon, France
4. Yap G-E, Tan A-H, Pang H-H (2007) Discovering and exploiting causal dependencies for robust mobile context-aware recommenders. IEEE Trans Knowl Data Eng 19:977–992
5. Yu Y, Kim J, Shin K, Jo GS (2009) Recommendation system using location-based ontology on wireless internet: an example of collective intelligence by using 'mashup' applications. Expert Syst Appl 36:11675–11681
6. Baltrunas L (2008) Exploiting contextual information in recommender systems. In: Presented at the proceedings of the 2008 ACM conference on recommender systems. Lausanne, Switzerland
7. Yeung KF, Yang Y, Ndzi D (2009) Contextualized mobile information retrieval in hybrid P2P environment. In: Presented at the 4th international conference on pervasive computing and applications, Taiwan
8. Cantador I, Bellogín A, Castells P (2008) Ontology-based personalised and context-aware recommendations of news items. In: Presented at the proceedings of the 2008 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology, vol 01
9. Das AS, Datar M, Garg A, Rajaram S (2007) Google news personalization: scalable online collaborative filtering. In: Presented at the proceedings of the 16th international conference on world wide web, Banff, Alberta, Canada
10. Liu D-R, Lai C-H, Lee W-J (2009) A hybrid of sequential rules and collaborative filtering for product recommendation. Inf Sci 179:3505–3519
11. Manouselis N, Costopoulou C (2007) Analysis and classification of multi-criteria recommender systems. World Wide Web 10:415–441
12. Adomavicius G, Sankaranarayanan R, Sen S, Tuzhilin A (2005) Incorporating contextual information in recommender systems using a multidimensional approach. ACM Trans Inf Syst 23:103–145
13. Zhang Y, Zhuang Y, Wu J, Zhang L (2009) Applying probabilistic latent semantic analysis to multi-criteria recommender system. AI Commun 22:97–107
14. Goldberg D, Nichols D, Oki BM, Terry D (1992) Using collaborative filtering to weave an information tapestry. Commun ACM 35:61–70
15. Lee HJ, Park SJ (2007) MONERS: a news recommender for the mobile web. Expert Syst Appl 32:143–150
16. Ono C, Kurokawa M, Motomura Y, Asoh H (2009) A context-aware movie preference model using a Bayesian network for recommendation and promotion, pp 247–257
17. Baudisch P, Brueckner L (2005) TV scout: lowering the entry barrier to personalized TV program recommendation, pp 299–309

18. Ziegler C-N, McNee SM, Konstan JA, Lausen G (2005) Improving recommendation lists through topic diversification. In: Presented at the proceedings of the 14th international conference on World Wide Web. Chiba, Japan
19. Liu D-R, Shih Y-Y (2005) Integrating AHP and data mining for product recommendation based on customer lifetime value. Inf Manag 42:387–400
20. Lee J, Lee J (2007) Context awareness by case-based reasoning in a music recommendation system, pp 45–58
21. Huang Y, Bian L (2009) A Bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the Internet. Expert Syst Appl 36:933–943
22. Burke R (2002) Hybrid recommender systems: survey and experiments. User Model User-Adapted Interact 12:331–370
23. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17:734–749
24. Chu W, Park S-T (2009) Personalized recommendation on dynamic content using predictive bilinear models. In: Presented at the proceedings of the 18th international conference on World wide web. Spain, Madrid
25. Chen A (2005) Context-aware collaborative filtering system: predicting the user's preference in the ubiquitous computing environment, pp 244–253
26. Yap G-E, Tan A-H, Pang H-H (2005) Dynamically-optimized context in recommender systems. In: Presented at the proceedings of the 6th international conference on mobile data management. Ayia Napa, Cyprus
27. Riva O, Toivonen S (2007) The DYNAMOS approach to support context-aware service provisioning in mobile environments. J Syst Softw 80:1956–1972
28. Baldauf M, Dustdar S, Rosenberg F (2007) A survey on context-aware systems. Int J Ad Hoc Ubiquitous Comput 2:263–277
29. Ruffo G, Schifanella R (2009) A peer-to-peer recommender system based on spontaneous affinities. ACM Trans Internet Technol 9: 1–34
30. Tennenhouse D (2000) Proactive computing. Commun ACM 43:43–50
31. Uchyigit G, Ma MY (2008) Personalization techniques and recommender systems. World Scientific Publishing Co. Pte. Ltd., Singapore
32. Kwon O, Choi S, Park G (2005) NAMA: a context-aware multi-agent based web service approach to proactive need identification for personalized reminder systems. Expert Syst Appl 29:17–32
33. Choeh JY, Lee HJ (2008) Mobile push personalization and user experience. AI Commun 21:185–193
34. Manouselis N, Costopoulou C (2007) Experimental analysis of design choices in multiattribute utility collaborative filtering. Int J Pattern Recogn Artif Intell 21:311–331
35. Kwon O, Kim M (2004) MyMessage: case-based reasoning and multicriteria decision making techniques for intelligent context-aware message filtering. Expert Syst Appl 27:467–480
36. JXTA (2011) JXTA. http://jxta.kenai.com/
37. Gu T, Pung HK, Zhang DQ (2005) A service-oriented middleware for building context-aware services. J Netw Comput Appl 28:1–18
38. Saaty TL (2008) Relative measurement and its generalization in decision making: why pairwise comparisons are central in mathematics for the measurement of intangible factors the analytic hierarchy/network process. Real Academia de Ciencias Exactas, Físicas y Naturales
39. Barzilai JJ (1997) Deriving weights from pairwise comparison matrices. J Oper Res Soc 48:1226–1232
40. Papadogiorgaki M, Papastathis V, Nidelkou E, Waddington S, Bratu B, Ribiere M, Kompatsiaris I (2008) Two-level automatic adaptation of a distributed user profile for personalized news content delivery. Int J Digit Multimedia Broadcasting 2008:21
41. Saaty TL (1980) The analytic hierarchy process : planning, priority setting, resource allocation. McGraw-Hill, New York
42. Forman E, Peniwati K (1998) Aggregating individual judgments and priorities with the analytic hierarchy process. Eur J Oper Res 108:165–169
43. Altuzarra A, Moreno-Jiménez JM, Salvador M (2007) A Bayesian priorization procedure for AHP-group decision making. Eur J Oper Res 182:367–382
44. Mikhailov L (2004) Group prioritization in the AHP by fuzzy preference programming method. Comput Oper Res 31:293–301
45. Ekel P, Queiroz J, Parreiras R, Palhares R (2009) Fuzzy set based models and methods of multicriteria group decision making. Nonlinear Anal Theory Methods Appl 71:e409–e419
46. Hong J, Suh E-H, Kim J, Kim S (2009) Context-aware system for proactive personalized service based on context history. Expert Syst Appl 36:7448–7457
47. Billsus D, Hilbert DM, Maynes-Aminzade D (2005) Improving proactive information systems. In: Presented at the proceedings of the 10th international conference on intelligent user interfaces. San Diego, California
48. Breese J, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: Presented at the proceedings of the fourteenth conference on uncertainty in artificial intelligence. Madison
49. Korb KB, Nicholson AE (2004) Bayesian artificial intelligence. Chapman & Hall/CRC, Boca Raton; London
50. Yeung KF, Yang Y (2010) A proactive personalized mobile news recommendation system. In: Presented at the DeSE 2010—developments in E-systems Engineering, London
51. Apache (2011) Apache Lucene project. http://lucene.apache.org/
52. Perst (2011) Perst—an open source, object-oriented embedded database. http://www.mcobject.com/perst
53. Liu J, Dolan P, Pedersen ER (2010) Personalized news recommendation based on click behavior. In: Presented at the proceeding of the 14th international conference on intelligent user interfaces. Hong Kong, China
54. Dutton WH, Helsper EJ, Gerber MM (2009) Oxford Internet Survey 2009 Report: The Internet in Britain. University of Oxford, Oxford Internet Institute
55. Mintel (2009) National Newspapers, UK
56. Billsus D, Pazzani MJ (2007) Adaptive news access. In: The adaptive web, vol. 4321/2007. Springer, Berlin, pp 550–570
57. Chu W, Park S-T, Beaupre T, Motgi N, Phadke A, Chakraborty S, Zachariah J (2009) A case study of behavior-driven conjoint analysis on Yahoo!: front page today module. In: presented at the proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining. France, Paris
58. Claypool M, Gokhale A, Miranda T, Murnikov P, Netes D, Sartin M (1999) Combining content-based and collaborative filters in an online newspaper. In: Presented at the proceedings of ACM SIGIR workshop on recommender systems
59. Billsus D, Pazzani MJ (2000) User modeling for adaptive news access. User Model User Adapted Interact 10:147–180
60. Antoniu G, Cudennec L, Jan M, Duigou M (2007) Performance scalability of the JXTA P2P framework. In: Parallel and distributed processing symposium IPDPS 2007. IEEE Int 2007:1–10
61. Satish Narayana S (2008) Scalable mobile web service discovery in peer to peer. Networks 668–674
62. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22:5–53