Knowl Inf Syst (2012) 31:345–370
DOI 10.1007/s10115-011-0417-1

REGULAR PAPER

An unsupervised approach to modeling personalized contexts of mobile users

Tengfei Bao · Huanhuan Cao · Enhong Chen · Jilei Tian · Hui Xiong

Received: 4 January 2011 / Revised: 30 March 2011 / Accepted: 11 May 2011 /
Published online: 28 May 2011

Abstract Mobile context modeling is a process of recognizing and reasoning about contexts and situations in a mobile environment, which is critical for the success of context-aware mobile services. While there are prior works on mobile context modeling, the use of unsupervised learning techniques for mobile context modeling is still under-explored. Indeed, unsupervised techniques have the ability to learn personalized contexts, which are difficult to be predefined. To that end, in this paper, we propose an unsupervised approach to modeling personalized contexts of mobile users. Along this line, we first segment the raw context data sequences of mobile users into context sessions where a context session contains a group of adjacent *context records* which are mutually similar and usually reflect the similar contexts. Then, we exploit two methods for mining personalized contexts from context sessions. The first method is to cluster context sessions and then to extract the frequent contextual feature-value pairs from context session clusters as contexts. The second method leverages topic models to learn personalized contexts in the form of probabilistic distributions of raw context data from the context sessions. Finally, experimental results on real-world data show that the proposed approach is efficient and effective for mining personalized contexts of mobile users.

Keywords Mobile context modeling · Unsupervised approach

T. Bao · E. Chen (✉)
University of Science and Technology of China, Hefei, Anhui, China
e-mail: cheneh@ustc.edu.cn

T. Bao · H. Cao · J. Tian
Nokia Research Center, Tampere, Finland

H. Xiong
Rutgers University, Newark, NJ, USA

1 Introduction

Recent years have witnessed a revolution in mobile devices, which is driven by the ever-increasing needs of mobile services. As mobile services keep evolving, there are clear signs that context modeling of mobile users will have huge demand. A distinct property of mobile users is that they are usually exposed in volatile contexts, such as waiting for a bus, walking in a building, driving a car, or doing shopping. Thus, building context-aware services by leveraging the rich contextual information of mobile users has attracted the great attention of many researchers [3,5,16,22].

Mobile context modeling is a process of recognizing and reasoning about contexts and situations in a mobile environment, which is a fundamental research problem toward leveraging the rich contextual information of mobile users. There are prior works on mobile context modeling such as [1,22]. However, most of these previous studies have a need to predefine the typical contexts of users and predetermine the corresponding rules for detecting them. While these approaches can work well in predefined simple application scenarios, such as guiding tourists for sightseeing [24], it is not flexible to extend these approaches for more general and complex scenarios where it is difficult to manually build context models. In addition, there are some other studies for mobile context modeling through supervised learning methods [17,27]. In this case, there is also a need to predefine contexts.

It is more attractive to exploit unsupervised techniques for mobile context modeling for the case that domain knowledge is not available, such as learning the personalized contexts that are difficult to be predefined. Indeed, unsupervised learning techniques can automatically learn some semantically meaningful contexts from the low level context data. In contrast, to model personalized contexts, both manual approach and supervised learning approach require users to predefine their own personalized contexts and thus will bring additional cost and complexity to the problem.

Therefore, in this paper, we propose an unsupervised approach to modeling personalized contexts of mobile users. Specifically, we first segment the raw context data sequence of mobile users into context sessions where a context session contains a group of adjacent context records that are mutually similar and may reflect the similar contexts. We use an adaptive segmentation approach named the minimum entropy segmentation [13] to address the challenges of context segmentation on determining the number of segments and the segmentation threshold. Then, we exploit two methods for mining personalized contexts from context sessions. The first method is to cluster context sessions and then to extract the frequent contextual feature-value pairs from context session clusters as contexts. The drawback of the clustering-based method is that it requires a context session that can only belong to one cluster, which corresponds to one context. The deterministic rule makes the results sensitive to wrong assignments. In contrast, topic models can solve the problem using a probabilistic distribution to representing whether a context session reflects a particular context. Therefore, the second method takes advantage of topic models to learn personalized contexts in the form of probabilistic distributions of raw context data from the context sessions. Due to the structural constraint of context sessions, the state-of-the-art topic models cannot directly apply to mobile context modeling. Therefore, we exploit to extend existing topic models for fitting mobile context modeling. We first extend a single-topic-based topic model named Mixture Unigram (MU) [19] to a mobile context model which assumes that each context session reflects one latent context. However, we observe that some context sessions may reflect multiple contexts because the context segmentation stage may not exactly detect all boundaries of context transitions. Based on this observation, we also extend a multiple-topic-based topic model named Latent Dirichlet Allocation (LDA) [6] for mobile context modeling. We conduct extensive experiments on the real-world mobile usage data. Experimental results show that the topic model-based method outperforms the clustering-based method in terms of the effectiveness of mining personalized contexts but less efficient than the latter in terms of the computational cost. Moreover, among the two topic models, the LDA-based model is more effective than that extended from MUC for mobile context modeling but less efficient than the latter in terms of the computational cost.

Overview. The rest of this paper is organized as follows. First, we briefly review some related works in Sect. 2. The basic idea of unsupervised mobile context modeling is introduced in Sect. 3. Then, the details of context segmentation are presented in Sect. 4. Followed in Sects. 5 and 6, we present the details of modeling personalized contexts of mobile users through clustering context sessions and topic models, respectively. In Sect. 7, we report the experimental results on the real-life history context data of users. Finally, we conclude this paper and pinpoint some future research directions in Sect. 8.

2 Related work

In general, the related work can be grouped into three categories. In the first category, contexts are modeled manually based on domain knowledge. For example, Schilit et al. [22] used key-value pairs to model the context by providing the value of a context information (e.g., location information) to an application as an environment variable. Adowd et al. [1] presented the Cyberguide project, in which prototypes of a mobile context-aware tour guide were built. Otzturk and Aamodt [20] proposed modeling the context with ontologies and analyzed psychological studies on the difference between recall and recognition of several issues in combination with contextual information. Indeed, none of the above-mentioned studies adopted machine learning approach for learning contexts from the raw context data automatically. As a result, they may work well in simple environments, such as guiding tourists in tourist attractions, but are not flexible for applying to more complex environments where it is difficult to build context models manually, e.g., recognizing users' contexts in their daily life.

The second category includes the research work of mobile context modeling though supervised learning approaches. For example, Liao et al. [17] attempted to infer an individual's transportation routine, given the user raw GPS data. By leveraging a dynamic Bayesian network, the system learns and infers the person's transportation routines between the significant places. Zheng et al. [27] exploited to use several supervised learning approaches for modeling user's raw GPS data. In their work, four different inference models including decision tree, Bayesian network, support vector machine (SVM), and conditional random field (CRF) are studied for modeling user's transportation mode. Supervised learning approach provides more flexibility than the manual approach for mobile context modeling because it depends on less domain knowledge and can learn from the raw context data automatically. However, it still needs to manually predefine the contexts. Moreover, it needs a number of labeled training data for model training. By contrast, the unsupervised learning approach for mobile context modeling is very flexible because it can learn contexts from an individual user's raw context data without predefined contexts nor labeled training data. Thus, it can greatly improve the user experience due to less dependency on the user.

The third category of related work focuses on user modeling through unsupervised approaches. In a latest literature, Eagle et al. [7] proposed to use the eigenvector of user behavior for modeling individual users and infer community affiliations within the subjects' social network. Though they also used an unsupervised approach to discover the user context and behavior pattern from the user history data, the objective of their research is intrinsically different from that of our work. Our goal is to discover the personalized mobile contexts that can be applied to context-aware services.

In addition, the proposed approach in this paper exploits topic models, which are widely used generative probability models in document modeling. Typical topic models include the Mixture Unigram (MU) model [19], the probabilistic Latent Semantic Analysis (pLSA) model [15], and the Latent Dirichlet Allocation (LDA) model [6]. Most of other topic models are extended from them and applied to specific applications. In our approach, we extend MU to MUC and extend LDA to LDAC for satisfying the constraint of context data.

3 Learning personalized mobile contexts from context logs

The context collection software on mobile devices can collect rich context data of mobile users through their personal context logs. A context log consists of a number of context records with timestamps, and a context record is formed as a group of raw context data, i.e., *contextual feature-value pairs*, where a contextual feature denotes a type of context data, such as *Day name*, *Speed*, and *Cell ID*, etc. The contextual value in a contextual feature-value pair indicates the value of the corresponding contextual feature at a particular time point. The context collection software can predefine a set of contextual features whose values should be collected, but a context record may miss the values of some contextual features because these values are not always available. For example, when a user is in door, the mobile device cannot receive the GPS signal. In context logs, only the contextual feature-value pairs whose contextual values are not missing are recorded.

From the contextual feature-value pairs in context logs, we may be able to discover some meaningful contexts of mobile users. For example, suppose Table 1 shows a part of the context log of Ada, we can see that in a workday and during time at AM8:00-9:00 , Ada's moving speed was high and the background was noisy observed by audio level, which might imply the context is that *she was driving a car to her work place*. Moreover, in a holiday during time at AM10:00-11:00, Ada was moving in door and the background is noisy. In addition, considering that the cell ID represents a shopping mall, the context might be that *Ada was go shopping*.

If several adjacent context records in a context log are mutually similar, we say that they make up a *context session*. The context records in the same context session may capture the similar context information of the mobile user. If two contextual feature-value pairs usually co-occur in same context sessions, they may represent the same context. An unsupervised approach can automatically discover the highly related contextual feature-value pairs that reflect the same context by taking advantage of their co-occurrences. Once a group of highly related contextual feature-value pairs are found, users can assign them meaningful context tags for binding them with multiple context-aware applications, such as context-aware reminder, context-aware recommendations. For example, if an unsupervised approach can discover that the contextual feature-value pairs (*Is a holiday?: Yes*), (*Time range: AM10:00-11:00*), (*Movement: Moving*), and (*Cell ID: 2552*) are highly related, Ada will be encouraged to tag this group of contextual feature-value pairs with an explicit context label "Go shopping" and define the services she wants on that context, such as playing a favorite music or recommending the information of fashion dress. This kind of semi-automatic context-aware configuration is more convenient than a manual alternative that lets Ada define the contextual feature-value pairs of "Go shopping" by herself. Along this line, we propose a two-stage

Table 1 A toy context log

Timestamp	Context record
t_1	{(Is a holiday?: No), (Time range: AM8:00–9:00), (Speed: High), (Audio level: Low), (Interaction: Listening music)}
t_2	{(Is a holiday?: No), (Time range: AM8:00–9:00), (Speed: High), (Audio level: Middle)}
t_3	{(Is a holiday?: No), (Time range: AM8:00–9:00), (Speed: High), (Audio level: Middle)}
	……
t_{38}	{(Is a holiday?: No), (Time range: AM10:00–11:00), (Movement: Not moving), (Audio level: Low), (Inactive time: Long)}
t_{39}	{(Is a holiday?: No), (Time range: AM10:00–11:00), (Movement: Not moving), (Audio level: Low), (Inactive time: Long)}
t_{40}	{(Is a holiday?: No), (Time range: AM10:00–11:00), (Movement: Not moving), (Audio level: Low), (Inactive time: Long)}
	……
t_{58}	{(Is a holiday?: Yes), (Time range: AM10:00–11:00), (Movement: Moving), (Cell ID: 2552), (Audio level: Middle)}
t_{59}	{(Is a holiday?: Yes), (Time range: AM10:00–11:00), (Movement: Moving), (Cell ID: 2552), (Audio level: High)}
t_{60}	{(Is a holiday?: Yes), (Time range: AM10:00-11:00), (Movement: Moving), (Cell ID: 2552), (Audio level: Middle)}

unsupervised approach for learning the personalized contexts of mobile users. In the first stage, we takes advantage of an adaptive segmentation approach to segment the context log into context sessions. In the second stage, we use the extended topic models to learn personalized contexts from the context sessions. The details of the approach are presented in the following sections.

4 Extracting context sessions

Given a context log $R = r_1 r_2 \ldots r_n$, where $r_i (1 \leq i \leq n)$ denotes a context record, extracting context sessions from R is a procedure of segmenting R into N segments $S = \{s_1, s_2, \ldots, s_N\}$, where $s_i (1 \leq i \leq N)$ denotes a context session that consists of a group of adjacent and similar context records, and S is called a N-segmentation of R.

There are two challenges for segmenting the context log into context sessions. First, it is hard to estimate the number of context sessions in a context log, i.e., the parameter N. It is because mobile users may have different frequencies of context transitions due to their life styles, which implies the numbers of context sessions in their personal context logs may also vary significantly. Therefore, the partition-based segmentation approach (e.g., [14,23]) cannot apply to context segmentation. Second, it is also difficult to define a unified similarity threshold to determine where the original context log should be segmented for each individual user's context log. Thus, the similarity threshold-based segmentation approach (e.g., [11,18]) cannot apply too. Though in the study by [2], Aggarwal proposed a segmentation framework that did not need to set the number of segments or the similarity threshold; however, this approach need to set the number of micro clusters and the number of nearest neighbors, which is also not applicable in our problem. To address the context segmentation problem, we need an adaptive approach that can automatically segment context logs according to their intrinsic statistic properties without external guidance.

Hermes et al. [13] proposed a minimum entropy approach that can segment pixels of an image adaptively without any domain knowledge-related parameter. The basic idea of the approach is to transform the objective of finding the optimized segmentation to finding the minimum conditional entropy of the pixels given the segmentation. This approach can be easily extended to segment context logs because context segmentation can also be transformed to the problem of seeking the minimum entropy. To be specific, if we measure the similarity between two adjacent context records through the probability that they are assigned into the same context session by a random segmentation, the objective of seeking the optimized segmentation becomes seeking the segmentation $S^* = \arg\max_S L(R|S)$, where $L(R|S)$ denotes the likelihood of all context records given the segmentation S. Seeking the maximum $L(R|S)$ is equal to seeking the maximum $logL(R|S)$. If we assume that 1) for each context record r, the probability to be assigned into a given context session s is independent, and 2) for each context feature-value pair p of a given context record r, the probability to be assigned into a given context session s is independent, $logL(R|S)$ can be expressed as follows.

$$\begin{aligned} logL(R|S) &= \sum_s \sum_{r_s} logP(r_s|S) \\ &= \sum_s \sum_{r_s} \sum_{p_{r_s}} logP(p_{r_s}|s) \\ &= \sum_s \sum_p n_{s,p} logP(p|s), \end{aligned} \tag{1}$$

where s denotes a context session in S, r_s denotes a context record in s, p_{r_s} denotes a contextual feature-value pair in r_s, p denotes a unique contextual feature-value pair, and $n_{s,p}$ indicates the occurrence number of the feature-value pair p in context session s. If we use $\frac{n_{s,p}}{N_p}$ to estimate $P(p,s)$, where N_p denotes the number of all feature-value pairs in R, Equation 1 can be transformed as follows.

$$(1) = N_p \sum_s \sum_p P(p,s) log(P(p|s)) = -N_p \cdot H(p|s),$$

where $H(p|s)$ denotes the conditional entropy of all contextual feature-value pairs given all context sessions. Therefore, the original problem is transformed to $S^* = \arg\min_S H(p|s)$.

Hermes et al. [13] have demonstrated that this problem can be addressed by taking advantage of the greedy optimization. To be specific, to search a N-segmentation with the minimum entropy, we first find a $N+1$-segmentation with the minimum entropy. Then, we try to merge each pair of adjacent context sessions and in this way find a N-segmentation $S^{'}$ with the minimum entropy, and $S^{'}$ is the exact solution of S^*. Moreover, $H(p|s)$ has a certain solution when N is equal to n. It is because in this case, there exists only one segmentation that each

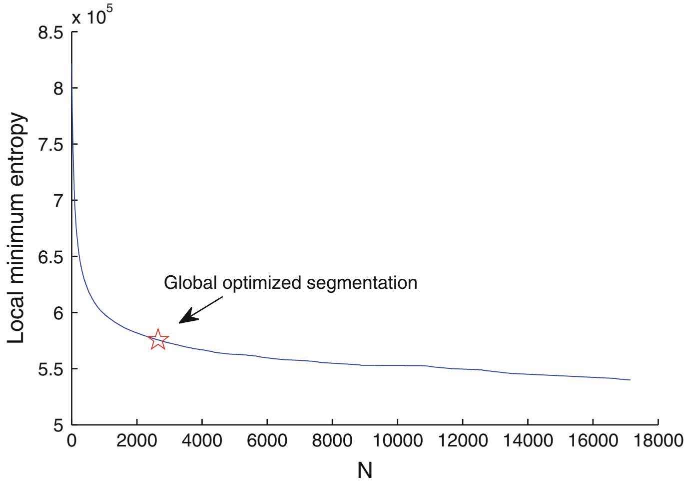


Fig. 1 Seeking the global optimized segmentation by balancing the complexity of the segmentation and the minimum entropy

context record makes up one context session. Therefore, we can easily find the optimized N-segmentation ($N \in [1, n]$).

It is easy to prove that the global minimum entropy appears when $N = n$ and the local minimum entropy given N increases with the decrease in N. However, only taking into account the minimum entropy usually causes over-fitting because such a segmentation is usually too complex. Therefore, we also take into account the growth rate of the local minimum entropy to balance the complexity of the segmentation and the corresponding local minimum entropy. To be specific, we start from $N = n$ and then iteratively set $N = N - 1$ and calculate the corresponding local minimum entropy. If the growth rate of the local minimum entropy is larger than ξ, we terminate seeking next local minimum entropy. Figure 1 illustrates the procedure of seeking the global optimized segmentation by balancing the complexity of the segmentation and the minimum entropy. The worst complexity of the adaptive segmentation approach is $O(NlogN)$.

5 Learning personalized contexts by clustering context sessions

In this section, we present a clustering-based method for learning personalized contexts by clustering context sessions. We assume that a group of similar context sessions may reflect the same context and can be used for extracting some key contextual feature-value pairs to represent the context. The basic idea is to cluster similar context sessions in terms of their contained raw contextual feature-value pairs and let a group of the most frequent contextual feature-value pairs in a cluster correspond a latent context. To be specific, we firstly propose an effective similarity measurement to calculate the similarity between context sessions. Then, we cluster the similar context sessions and extract contexts from the clusters.

5.1 Similarity between context sessions

Intuitively, we can assume that two context sessions are similar if they have similar distributions of contextual feature-value pairs. To this end, we firstly represent a context session as a L_2-normalized vector of contextual feature-value pairs or *CFVP-vector* for short. A CFVP-vector has P-dimensions where P indicates the total number of all unique contextual feature-value pairs appearing in the user context log. Precisely, the j-th element of a CFVP-vector $\overrightarrow{s_i}$ is

$$s_{i,j} = \begin{cases} Norm(freq_{i,j}) & \text{if } p_j \in s_i; \\ 0 & \text{otherwise}, \end{cases} \quad (2)$$

where p_j denotes the j-th unique contextual feature-value pair, $freq_{i,j}$ indicates the frequency of p_j in s_i, and $Norm(freq_{i,j}) = \frac{freq_{i,j}}{\sqrt{\sum_k freq_{i,k}^2}}$.

The similarity between two context sessions s_{i_1} and s_{i_2} is measured by the Euclidean distance between their CFVP-vectors [10]. That is,

$$Distance(\overrightarrow{s_{i_1}}, \overrightarrow{s_{i_2}}) = \sqrt{\sum_{j=1}^{P} (s_{i_1,j} - s_{i_2,j})^2}, \quad (3)$$

5.2 Clustering context sessions

With the similarity measure of context sessions, we can cluster context sessions and let the most frequent contextual feature-value pairs of a cluster correspond to a context. Since the proposed similarity measure is in a form of distance function of two vectors, we need a spatial clustering algorithm. As is well known, the existing spatial clustering algorithms can be classified into three categories: partition-based clustering algorithms (e.g., K-means), density-based clustering algorithms (e.g., DBSCAN [8]), and stream-based clustering algorithms (e.g., BIRCH [26]). Both the density-based clustering algorithms and the stream-based clustering algorithms need a predefined parameter to control the granularity of the clusters. Because the properties of different contexts are volatile, the granularity of different context session clusters may be diverse due to the different numbers of occurrences for each context. For example, a context that *the user is working in the office* usually appears every day and covers many related context sessions, while another context that *the user is having a drink in a pub* may only appear in week ends and thus cover much less context sessions. Therefore, it may be infeasible to control the granularity of all clusters by the same predefined parameter. As a consequence, we use partition-based clustering algorithms for clustering contextual feature-value pairs. To be specific, we use the well-known K-means clustering algorithm.

The details of clustering contextual feature-value pairs by K-means are described as follows. We firstly randomly select K context sessions as the mean nodes of K clusters and assign other context sessions to the K clusters according to their distances to the mean nodes. Then, we iteratively calculate the mean of each cluster and reassign the context sessions until the assignment does not change or the iteration exceeds to the max number of iterations. Algorithm 1 shows the pseudocode of clustering context sessions by K-means, where $L^t = L^{t-1}$ means $\forall_i(l_i^t = l_i^{t-1})$ and N_k^t indicates the number of context sessions with label k in the t-th iteration.

5.3 Extracting contexts from context session clusters

A context session cluster contains several similar context sessions that may reflect the same context. Therefore, we can use a context session cluster as the source for extracting a personalized context of the mobile user. To be specific, we extract the contextual feature-value pairs whose frequency is larger than a predefined threshold from a context session cluster. In

Algorithm 1 Clustering Contextual Feature-value Pairs by K-means

Input:
1) a set of context sessions in the form of CFVP-vectors $\overrightarrow{S} = \{\overrightarrow{s_i}\}$;
2) the number of target clusters K;
3) the max number of iterations T;
Output: a set of cluster labels $L = \{l_i\}$, where l_i is the label of $\overrightarrow{s_i}$;
Init: Create a set of cluster labels $L^0 = \{l_i^0\}$, where l_i^0 corresponds to $\overrightarrow{p_i}$ and is a random integer selected from $[1, K]$;

1: randomly choose K context sessions as mean nodes, denoted as $\overrightarrow{s^0_{i_1}}, \overrightarrow{s^0_{i_2}}, \ldots, \overrightarrow{s^0_{i_K}}$;
2: **for** $t = 1; t < T; ++t$ **do**
3: create a set of cluster labels $L^t = \{l_i^t\}$, where l_i^t is a random integer selected from $[1, K]$;
4: **for each** $\overrightarrow{s_i}$ **in** $\overrightarrow{S}$ **do**
5: $l_i^t = \arg\min_k Distance(\overrightarrow{s_i}, \overrightarrow{s^{t-1}_{i_k}})$, where $1 \le k \le K$;
6: **if** L^t=L^{t-1} **then**
7: go to 11;
8: **else**
9: **for** $k = 1; k < K; k++$ **do**
10: $\overrightarrow{s^t_{i_k}} = \frac{\sum_{l_i^t = k} \overrightarrow{s_i}}{N_k^t}$;
11: return L^t.

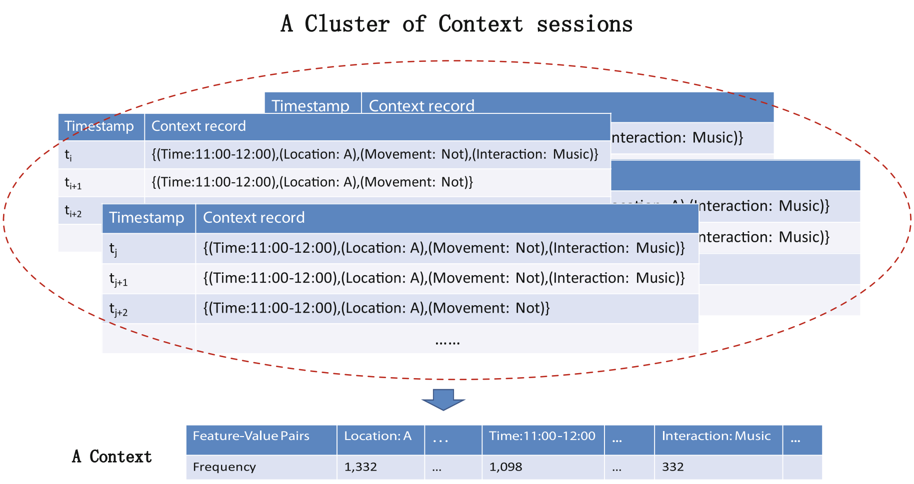


Fig. 2 An example of extracting contexts from context session clusters

practice, the threshold is set to be $N_k \cdot 0.01$, where N_k indicates the total number of contextual feature-value pairs in the context session cluster c_k. Figure 2 shows an example of extracting contexts from context session clusters.

5.4 Determining the number of context session clusters

The partition-based clustering algorithms need a predefined parameter K that indicates the number of target clusters. The selection of K is essentially equal to the selection of the number of contexts because each context session cluster is used to extract one context. Thus, to select an appropriate K, we can assume that the number of contexts for mobile users must fall into a range $[K_{min}, K_{max}]$, where K_{min} and K_{max} indicate the minimum number and the maximum number of possible contexts, respectively. The values of $K_{\min}$ and $K_{\max}$ can be empirically determined through the user study that selects users with different backgrounds and asks them how many typical contexts exist in their daily life. Thus, we can select the best K from $[K_{\min}, K_{\max}]$ by measuring the clustering performance for a specific user's context data set.

Since there is no ground truth for clusters, we evaluate the clustering performance indirectly by evaluating the quality of learned contexts for modeling context data set. To be specific, we firstly partition a context session set S into a training set S_a and a test set S_b. Then, we run K-means on S_a with a given K and obtain K clusters of context sessions as K contexts $c_1, c_2, \ldots, c_K$. Then, we calculate the perplexity [4] of the S_b by the following equation.

$$Perplexity(S_b) = Exp\left[-\frac{\sum_{s\in S_b} log P(s|S_a)}{\sum_{s\in S_b} N_s}\right], \tag{4}$$

where s denotes a context session, $P(s|S_a)$ means the probability that s occurs given S_a, and N_s indicates the number of contextual feature-value pairs in s.

In the clustering-based context model, $P(s|S_a)$ is calculated as follow.

$$\begin{aligned} P(s|S_a) &= \prod_{p_i\in s} P(p_i|S_a) = \prod_{p_i\in s}\sum_{c_k} P(p_i, c_k|S_a) \\ &= \prod_{p_i\in s}\sum_{c_k} P(p_i|c_k)P(c_k|S_a) \end{aligned}$$

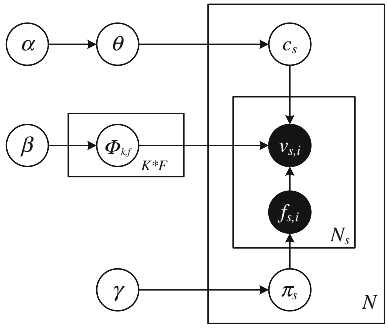
where p_i denotes a contextual feature-value pair of s, c_k denotes a cluster context sessions. $P(p_i|c_k)$ is calculated as $\frac{freq_{i,k}}{N_k}$, where $freq_{i,k}$ indicates the frequency of p_i in c_k and N_k indicates the total number of contextual feature-value pairs in c_k. $P(c_k|S_a)$ is calculated as $\frac{|c_k|}{N}$, where $|c_k|$ indicates the number of contest sessions in c_k and N indicates the total number of context sessions. The smaller the perplexity is, the better the learned contexts' quality will be.

It is worth noting that we observe the perplexity of K-means roughly drops with the increase of K in the experiments. If we only take into account the perplexity, we probably select the maximum K of a given range, which may make the learned model over-fitting. Thus, we balance the above-mentioned approach by a simple way; that is, if the reducing ratio of perplexity is less than τ, we do not select a larger K. In practice, we set τ to be 10% according to experiment analysis.

6 Learning personalized contexts by topic models

The drawback of the clustering-based method for learning personalized contexts is that it requires a context session that can only belong to one cluster, which corresponds to one context. The deterministic rule makes the results sensitive to wrong assignments. In contrast, topic models can solve the problem by using a probabilistic distribution to represent whether a context session reflects a particular context. Topic models are generative models that are successfully used for document modeling. They assume that there exist several topics for a corpus D and a document d in D can be taken as a bag of words $\{w_{d,i}\}$, which are generated by these topics. Intuitively, if we take contextual feature-value pairs as words, take context sessions as bags of contextual feature-value pairs to correspond documents and take latent

Fig. 3 The graphical representation of the MUC model



contexts as topics, we can take advantage of topic models to learn contexts from context sessions. However, we cannot directly apply topic models to mobile context modeling because the occurrences of the contextual features and the corresponding values in contextual feature-value pairs are dependent on different factors. As mentioned earlier, in a context session, the occurrences of contextual features are dependent on some external conditions, such as the availability of the corresponding signal. In contrast, the occurrences of contextual values are dependent on the latent contexts and the corresponding contextual features. If we simply take contextual feature-value pairs as words in topic models, we will not be able to discriminate the generation of contextual features and that of contextual values. To this end, we extend the existing topic models for fitting mobile context modeling.

6.1 Single-context-based context model

If we assume that one context session reflects one latent context, we can extend a typical singe-context-based topic model named the Mixture Unigram (MU) model [19] for mobile context modeling. MU assumes that a document d is generated as follows. Given K topics and M words, to generate the word $w_{d,i}$ in d, the model firstly generates a topic z_d from a prior topic distribution for the corpus D. Then, the model generates $w_{d,i}$ given the prior word distribution for z_d. In a corpus, both the prior topic distribution and the prior word distributions for different topics follow the Dirichlet distribution.

We extend the MU model to the Mixture Unigram on Context (MUC) model which assumes that a context session is generated by a prior contextual feature distribution and a prior context distribution together. To be specific, given K contexts and F contextual features, the MUC model assumes that a context session s is generated as follows. Firstly, a global prior context distribution θ is generated from a prior Dirichlet distribution α. Secondly, a prior contextual feature distribution π_s is generated from a prior Dirichlet distribution γ. Then, a context c_s is generated from θ. Finally, a contextual feature $f_{s,i}$ is generated from π_s, and the value of $f_{s,i}$ denoted as $v_{s,i}$ is generated from the distribution $\phi_{c_s, f_{s,i}}$. Moreover, there are totally $K \times F$ conditional distributions of contextual feature-value pairs $\{\phi_{k,f}\}$ that follow a Dirichlet distribution β. Figure 3 shows the graphical representation of the MUC model. Notice that α, β and γ are represented by parameter vectors $\overrightarrow{\alpha} = \{\alpha_k\}$, $\overrightarrow{\beta} = \{\beta_v\}$, and $\overrightarrow{\gamma} = \{\gamma_f\}$, respectively, according to the definition of Dirichlet distribution, where k indicates a context, v indicates a contextual value, and f indicates a contextual feature.

In the MUC model, given the parameters α, β, and γ, the joint probability of a context session $s = \{(f_{s,i} : v_{s,i})\}$, a prior context distribution θ, a latent context c_s, a contextual feature distribution π_s, and a set of $K \times F$ conditional contextual value distributions $\Phi = \{\phi_{k,f}\}$

is calculated as follows.

$$P(s, \theta, c_s, \pi_s, \Phi|\alpha, \beta, \gamma) = P(c_s|\theta)P(\theta|\alpha)P(\Phi|\beta)P(\pi_s|\gamma) \times \left(\prod_{i=1}^{N_s} P(v_{s,i}|c_s, f_{s,i}, \Phi)P(f_{s,i}|\pi_s)\right),$$

where $P(v_{s,i}|c_s, f_{s,i}, \Phi) = P(v_{s,i}|c_s, f_{s,i}, \phi_{c_s, f_{s,i}})$ and N_s indicates the number of contextual feature-value pairs in s.

The likelihood of a set of context sessions S is calculated as follows.

$$\begin{aligned} L(S) &= \prod_s P(s|\alpha, \beta, \gamma) \\ &= \prod_s \int\int\int \left(\sum_{c_s} \prod_{i=1}^{N_s} P(p_{s,i}|c_s, f_{s,i}, \Phi)P(f_{s,i}|\pi_s)P(c_s|\theta)\right) \\ &\quad \times P(\theta|\alpha)P(\Phi|\beta)P(\pi_s|\gamma)d\theta d\Phi d\pi_s, \end{aligned}$$

The representation of the likelihood of MUC is in a too complex form and it may not be feasible to calculate the parameters of the model directly. Alternatively, we use a commonly used iterative approach for approximately estimating the parameters of MU called Gibbs sampling [12,21]. In the Gibbs sampling approach, each observed variable is iteratively assigned a label by taking into account the labels of other observed variables. For our problem, the Dirichlet parameter vectors $\overrightarrow{\alpha}$, $\overrightarrow{\beta}$, and $\overrightarrow{\gamma}$ are empirically predefined first. Then the Gibbs sampling approach iteratively assigns context labels to each context session according to the labels of other context sessions.

The Gibbs sampler of the context label for a context session s, denoted as c_s, is defined as follows.

$$\begin{aligned} P(c_s = k|C_{\neg s}, S) &\propto P(c_s = k, C_{\neg s}, S) \\ &= P(c_s = k, C_{\neg s}, V, F) \\ &= P(V|c_s = k, C_{\neg s}, F)P(c_s = k|C_{\neg s})P(F) \\ &\propto P(v_s|c_s = k, C_{\neg s}, F, V_{\neg s})P(c_s = k|C_{\neg s}), \end{aligned}$$

where $\neg s$ means removing s from S, $C_{\neg s}$ denotes the context labels of other context sessions expect for s, V and F denote all contextual values and all contextual features in S, respectively, and v_s denotes all contextual values in s.

Moreover, indicating the token (s, i) as m, we have the following formulas.

$$\begin{aligned} P(v_s|c_s = k, C_{\neg s}, F, V_{\neg s}) &= \prod_{i=1}^{N_s} P(v_{m=(s,i)}|c_s = k, C_{\neg s}, F, V_{\neg s}) \\ &= \prod_{i=1}^{N_s} \frac{n_{\neg s,k,f_m,v_m} + \beta_{v_m}}{\sum_v n_{\neg s,k,f_m,v} + \sum_{v \in V_{f_m}} \beta_v} \\ P(c_s = k|C_{\neg s}) &= \frac{n_{\neg s,k} + \alpha_k}{N - 1 + \sum_{k'=1}^{K} \alpha_{k'}}, \end{aligned}$$

where $n_{\neg s,k,f,v}$ indicates the frequency that the contextual feature-value pair $(f : v)$ is labeled with the k-th context in all context sessions expect for s, V_f denotes the set of contextual values for the contextual feature f, and $n_{\neg s,k}$ indicates the number of context sessions with the k-th context expect for s.

After several rounds of Gibbs sampling, eventually each context session will be assigned a final context label. We can derive the personalized contexts of mobile users from the labeled context sessions by estimating the probability distribution of contextual feature-value pairs generated by a particular context. To be specific, the probability that a contextual feature-value pair $p_m = (f_m : v_m)$ is generated by the context c_k is estimated as

$$P(p_m|c_k) = P(v_m|c_k, f_m)P(f_m), \tag{5}$$

where

$$P(v_m|c_k, f_m) = \frac{n_{k,f_m,v_m} + \beta_{v_m}}{\sum_v n_{k,f_m,v} + \sum_{v \in V_{f_m}} \beta_v}$$

$$P(f_m) = \frac{\sum_{k'=1}^{K} \sum_v n_{k',f_m,v} + \gamma_{f_m}}{\sum_f \sum_{k'=1}^{K} \sum_v n_{k',f_m,v} + \sum_f \gamma_f}.$$

6.2 Multiple-context-based context model

In practice, the stage of context segmentation may not detect the exact boundaries of context sessions. Therefore, it is more general to assume that one context session may reflect multiple latent contexts. To this end, we also propose a multiple-context-based context model which is extended from a multiple-topic-based topic model named the Latent Dirichlet Allocation (LDA) model [6]. Compared with MU, LDA assumes each document is generated by a prior distribution of topics instead of a single topic. To be specific, LDA assumes that a document d is generated as follows. Given K topics and M words, to generate the word $w_{d,i}$ in d, the model firstly generates a topic $z_{d,i}$ from a prior topic distribution for d. Then the model generates $w_{d,i}$ given the prior word distribution for $z_{d,i}$. Moreover, similar to MU, LDA assumes that both the prior topic distributions for different documents and the prior word distributions for different topics follow the Dirichlet distribution.

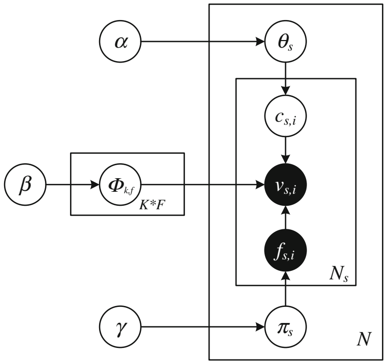
We extend LDA to the Latent Dirichlet Allocation on Context model (LDAC) for mobile context modeling. In the LDAC model, a context session s is generated as follows. Firstly, a prior context distribution θ_s is generated from a prior Dirichlet distribution α. Secondly, a prior contextual feature distribution π_s is generated from a prior Dirichlet distribution γ. Then, for the i-th contextual feature-value pair in s, a context $c_{s,i}$ is generated from θ_s, a contextual feature $f_{s,i}$ is generated from π_s, and the value of $f_{s,i}$ denoted as $v_{s,i}$ is generated from the distribution $\phi_{c_{s,i},f_{s,i}}$. Moreover, there are totally $K \times F$ prior distributions of contextual feature-value pairs $\{\phi_{k,f}\}$ which follow a Dirichlet distribution β. Figure 4 shows the graphical representation of the LDAC model.

In the LDAC model, given the parameters α, β and γ, the joint probability of a context session $s = \{(f_{s,i} : v_{s,i})\}$, a prior context distribution θ_s, a group of latent context labels $c_s = \{c_{s,i}\}$, a contextual feature distribution π_s, and a set of $K \times F$ conditional contextual value distributions $\Phi = \{\phi_{k,f}\}$ is calculated as follows.

$$P(s, \theta, c_s, \pi_s, \Phi|\alpha, \beta, \gamma) = P(\theta_s|\alpha)P(\Phi|\beta)P(\pi_s|\gamma) \times \left(\prod_{i=1}^{N_s} P(v_{s,i}|c_{s,i}, f_{s,i}, \Phi)P(f_{s,i}|\pi_s)P(c_{s,i}|\theta_s)\right).$$

Similar to the parameter estimation in MUC, we also use the Gibbs sampling approach to estimating the parameters for LDAC. Denoting the token (s, i) as m, the Gibbs sampler of

Fig. 4 The graphical representation of the LDAC model



c_m is as follows.

$$
\begin{aligned}
P(c_m = k|C_{\neg m}, S) \propto\ & P(c_m = k, C_{\neg m}, S) \\
\propto\ & P(v_m|c_m = k, C_{\neg m}, F, V_{\neg m}) \\
& \times P(c_m = k|C_{\neg m}),
\end{aligned}
$$

where $\neg m$ means removing the contextual feature-value pair $(f_m : v_m)$ from S, and

$$
\begin{aligned}
P(v_m|c_m = k, C_{\neg m}, F, V_{\neg m}) &= \frac{n_{\neg m,k,f_m,v_m} + \beta_{v_m}}{\sum_v n_{\neg m,k,f_m,v} + \sum_{v \in V_{f_m}} \beta_v} \\
P(c_m = k|C_{\neg m}) &= \frac{n_{s,\neg m,k} + \alpha_k}{\sum_{k'=1}^{K} n_{s,\neg m,k'} + \sum_{k'=1}^{K} \alpha_{k'}},
\end{aligned}
$$

where $n_{\neg m,k,f,v}$ indicates the frequency that the contextual feature-value pair $(f : v)$ is labeled with the k-th context in all context sessions after removing the m-th contextual feature-value pair, and $n_{s,\neg m,k}$ indicates the number of contextual feature-value pairs labeled with the k-th context in s expect for the m-th one.

Similar to MUC, in the LDAC model, the personalized contexts of mobile users can also be derived from the labeled contextual feature-value pairs according to Equation 5. From the experimental results on real data, we find that LDAC outperforms MUC with respect to the effectiveness of mobile context modeling. However, the effectiveness of MUC is also acceptable, and it largely outperforms LDAC in terms of efficiency. Generally, MUC is a good candidate approach to mobile context modeling when the computation resource is limited. Otherwise, we can use LDAC for pursuing the best performance. The detailed comparisons of the practical performance between MUC and LDAC are presented in Sect. 7.

6.3 Determining the number of contexts

Both of MUC and LDAC need a predefined parameter K to indicate the number of contexts to be learned. Similar to what we do for the clustering-based method of learning contexts, to select an appropriate K, we assume that the number of personalized contexts for any mobile user falls into a range $[K_{\min}, K_{\max}]$, where $K_{\min}$ and $K_{\max}$ indicate the minimum number and the maximum number of possible contexts, respectively. Thus, we can select the best K from $[K_{\min}, K_{\max}]$ by measuring the performance of the learned context models. To be specific, we first partition a context session set S into a training set S_a and a test set S_b. Then, we learn a context model from S_a with a given K and obtain K contexts $c_1, c_2, \ldots, c_K$. Last,

Table 2 The details of the Reality Mining data sets used in our experiments

Owner ID	n	N	P	N_p
1	23,114	7,029	1,702	188,342
2	26,157	7,170	1,950	204,902
3	9,115	2,712	1,209	72,225
4	14,588	3,487	690	112,689
5	16,544	5,144	1,024	133,500
6	21,011	4,940	1,740	168,228
7	16,225	4,762	1,623	130,404
8	26,352	6,136	1,024	204,760
9	10,592	2,587	954	82,957
10	30,955	4,245	2,326	241,620

we calculate the perplexity Equation 4, where $P(s|S_a)$ is calculated as follows.

$$P(s|S_a) = \prod_{p_m \in s} P(p_m|S_a) = \prod_{p_m \in s} \sum_{k=1}^{K} P(p_m, c_k|S_a)$$
$$= \prod_{p_m \in s} \sum_{k=1}^{K} P(p_m|c_k, S_a) P(c_k|S_a),$$

where $P(p_m|c_k, S_a) = P(p_m|c_k)$ can be calculated by Equation 5 and $P(c_k|S_a)$ is calculated differently in MUC and LDAC. In the MUC model, $P(c_k|S_a) = \frac{n_k + \alpha_k}{N + \sum_{k'=1}^{K} \alpha_{k'}}$, where n_k indicates the number of context sessions labeled with c_k in S_a. In the LDAC model, $P(c_k|S_a) = \frac{n_{s,k} + \alpha_k}{\sum_{k'=1}^{K} n_{s,k'} + \alpha_{k'}}$, where $n_{s,k}$ indicates the number of contextual feature-value pairs labeled with c_k in s.

Similar to the clustering-based method for learning contexts, if we only take into account the perplexity, we probably select the maximum K of a given range, which may make the learned model over-fitting. Thus, we balance the above-mentioned approach by setting a threshold of the perplexity decline rate τ.

7 Experiments

In this section, we evaluate the efficiency and the effectiveness of the proposed approach for mobile context modeling through extensive experiments on real context data sets.

7.1 Data sets and preprocess

The first data set used in the experiments is the Reality Mining data set [7]. Reality Mining data set is a public data set that captures the raw context data from 100 college volunteers at MIT over the course of the 2004-2005 academic year. The raw context data contain the communication, proximity, location, and activity information and can be used for learning personalized contexts of the users. We randomly select 10 volunteers' context data from the Reality Mining data set to evaluate the performance of the proposed approach of mobile context modeling. Table 2 lists the details of the Reality Mining data sets used in our experiments, where the *Owner ID* identifies the owner of the context data, n denotes the number of

Table 3 The collected contextual features in Rich Context

Contextual feature	Value range
Day name	{Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday}
Is a holiday?	{True, False}
Day period	{Morning(AM7:00-AM11:00), Noon(AM11:00-PM14:00), Afternoon(PM14:00-PM18:00), Evening(PM18:00-PM21:00), Night(PM21:00-Next day AM7:00)}
Time range	{AM0:00-AM1:00, AM1:00-AM2:00, AM2:00-AM3:00, …, PM23:00-PM24:00}
Profile type	{General, Silent, Meeting, Outdoor, Pager, Offline}
Battery level	{Low(<25%), Middle(25%–50%), High(50%–80%), Full(>80%)}
Inactive time	{Short(<5 minutes), Middle(5–30 minutes), Long(>30 minutes)}
Ring type	{Normal, Ascending, Ring once, Beep, Silent}.
Cell ID	Integers.
Area ID	Integers.
Speed	{Low(<5 km/h), Middle(5–20 km/h), High(>20 km/h)}
Movement	{Moving, Not moving}
Coordinate	Pair of longitude and latitude.
Application	{Call, Message, Web browsing, Music, Video, E-book, Radio, Game}

context records, N denotes the number of extracted context sessions, P denotes the number of unique contextual feature-value pairs, and N_p denotes the occurrence number of all contextual feature-value pairs. In experiments, we find that the setting of ξ does not influence the segmentation significantly in a wide range. Explicitly, we set ξ to be 10%.

The evaluation of unsupervised approaches is challenging because of the lack of ground truth. Though some metrics such as perplexity can be applied to evaluating the proposed approach, it is more desirable to ask users to manually evaluate the personalized contexts learned from their raw context data. However, it is difficult to contact the owners of the reality mining data sets and ask them to conduct manual evaluations. To this end, we collect 10 college volunteers' context data spanning for one month through their mobile devices by ourselves. The collected context data set includes rich types of contextual features listed in Table 3, and the owners of these context data are invited to participate the human evaluation of the proposed approach to mobile context modeling. For simplicity, we denote the collected context data set as *Rich Context*. The details of Rich Context data sets are listed in Table 4.

We first partition each experimental data set into the training set and the test set as follows. For each Reality Mining data set, we use the last month data as the test set and use the remaining data as the training set. For each Rich Context data set, we use the last week data as the test set and use the remaining data as the training set. Then, we use the proposed approach

Table 4 The details of the Rich Context data sets

Owner ID	n	N	P	N_p
1	29,910	6,403	990	369,691
2	19,959	4,006	1,143	250,848
3	29,587	5,633	702	361,783
4	35,979	6,071	509	448,187
5	17,149	2,231	499	213,623
6	26,461	4,976	1,044	326,096
7	25,642	4,222	366	314,968
8	38,664	7,476	1,475	483,116
9	13,977	2,652	330	173,822
10	19,422	3,910	374	240,263

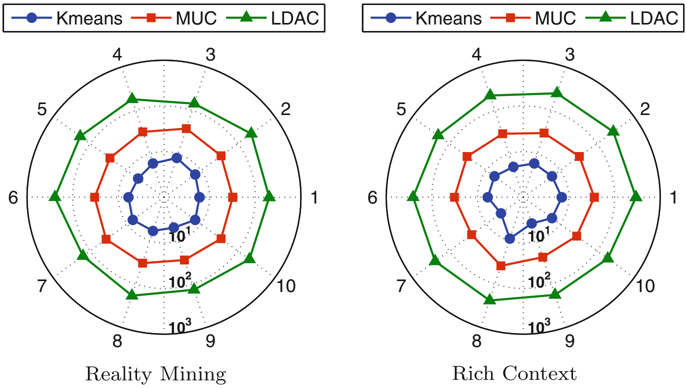


Fig. 5 Spherical comparison in terms of the request iterations to converge

to learn mobile contexts from each training set and then evaluate the learned contexts on the corresponding test set.

7.2 Efficiency of the proposed approaches

For the sake of privacy concern, one simple alternative solution is to model the personalized contexts of mobile users in their mobile devices instead of in a back end server. Thus, the efficiency of mobile context modeling is crucial for in-device applications due to the resource constraint of mobile device. In the experiments, we observe that the computation cost of extracting context sessions is trivial compared with that of learning contexts by topic models (averagely less than 20 seconds). Thus, we evaluate the efficiency of the proposed approach by comparing the efficiencies of K-means, MUC, and LDAC for mobile context modeling. Since all of the approaches adopt iterative learning methods, we evaluate their efficiencies by taking into account their convergence speeds. The experiments are conducted on a Core2 1.86GZ, 2G memory PC.

The convergence of K-means is measured by the ratio of reassigned context labels in one iteration, and the convergence of Gibbs sampling is measured by the log likelihood of the training set. The super parameters α, β, and γ of MUCs and LDACs are empirically set to $50/K$, 0.01, and 0.01 according to [12], where K indicates the number of contexts. Figure 5

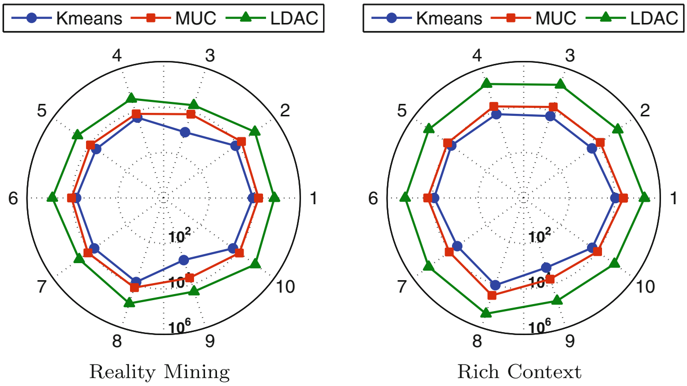


Fig. 6 Spherical comparison in terms of the time cost to converge (ms)

compares the request iterations with converge for K-means, MUC, and LDAC on the Reality Mining data set and the Rich Context data set, respectively. Each label around the circle indicates the owner ID of a data set. For each data set, the most appropriate K is selected by the method mentioned in Sect. 5. From this figure, we can see that the Gibbs sampling process of LDAC usually converges after hundreds of iterations while that of MUC usually converges after less than 30 iterations, and K-means usually converges after less than 10 iterations.

Figure 6 further compares the time cost to converge for K-means, MUC and LDAC. From this figure we can see that the time cost to converge for K-means is lowest and the Gibbs sampling process of MUC usually converges tens of times faster than that of LDAC. In a summary, though all of the proposed methods can converge within limited iterations, K-means is much more efficient than the two topic models. Moreover, MUC is much more efficient than LDAC for learning personalized mobile contexts. It is because the Gibbs sampling for LDAC is more complex than that of MUC. To train a LDAC model, we need to build a Gibbs sampler for each contextual feature-value pair. In contrast, the training of MUC only needs to build Gibbs samplers for context sessions, which are much fewer than contextual feature-value pairs in practice. Consequently, the Gibbs sampling of MUC largely outperforms that of LDAC in terms of both the time cost of one iteration and the iterations to converge.

7.3 Effectiveness of the proposed approaches

In this section, we report the experimental results of the proposed approach with respect to the effectiveness for mobile context modeling.

7.3.1 Perplexity

Figure 7 compares the perplexity of each test set with the contexts learned by K-means, MUC, and LDAC. Each label around the circle indicates the owner ID of a test set. From this figure, we can see that the two topic models always outperform K-means and LDAC always outperforms MUC in terms of perplexity, which concludes that LDAC is the most effective method for mobile context modeling.

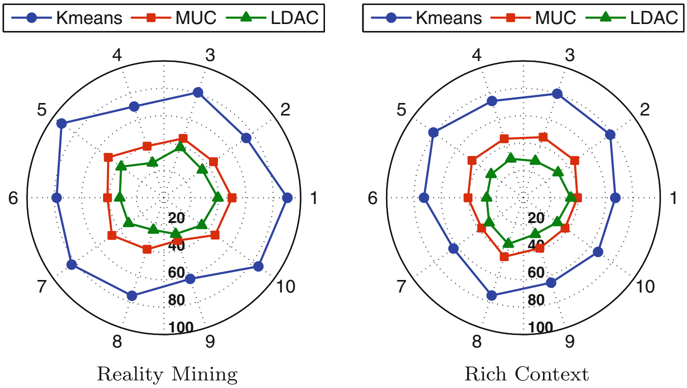


Fig. 7 Spherical comparison in terms of perplexity

7.3.2 Human evaluation

To find out the quality of the learned contexts more intuitively, we ask the owners of the Rich Context data sets to evaluate the personalized contexts learned from their own context data. For each learned context, we select the contextual feature-value pairs p where $P(p|c_k) > 0.01$ to represent the context c_k.

For each learned context to be evaluated, the corresponding testee selects one from the following three remarks:

- P: **P**erfect. This remark means that the learned context reflects one of the testee's typical contexts well. No irrelevant context information is included, and no relevant context information is missing.
- G: **G**ood. This remark means that the learned context partially reflects one of the testee's typical contexts but contains some irrelevant context information or misses some relevant information.
- B: **B**ad. This remark means that it is hard to state the learned context reflects which typical context of testee.

To ease of the evaluation, we leverage Google map to show the related locations of each context, and thus, the testees can conventionally bridge a context to their daily lives through the intuitive way. Figure 8 shows an example of context map.

To ensure the evaluation quality, we do not inform testees that a given learned context is learned by which context model. Moreover, we generate a copy for each learned context and randomly mix them with the original learned contexts. If a learned context pattern is assigned different remarks from that of its copy, we will revisit it again. Figure 9 compares the human evaluation results of the contexts learned by K-means, MUC, and LDAC for each data set of Rich Context. From the figure, we can see that the two topic models always outperform K-means in terms of perfect cases and good cases. Moreover, LDAC outperforms MUC for mobile context modeling in terms of perfect cases. But considering all positive cases (P+G), their performance are comparable. Generally speaking, we can conclude that the two topic models always outperform K-means and LDAC outperforms MUC in terms of effectiveness for mobile context modeling, which is consistent with the experimental conclusion in the view of perplexity.

Fig. 8 An example of context map for ease of evaluating contexts

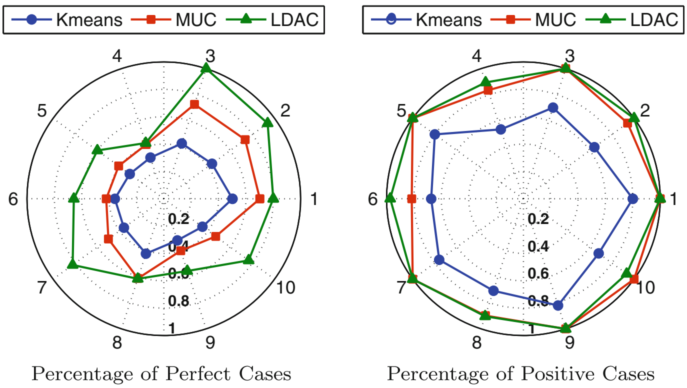


Fig. 9 Spherical comparison in terms of human evaluation for Rich Context data

Table 5 Context c_a learned by K-means

(Is a holiday?: No)
(Day name: Monday)
(Day name: Tuesday)
(Day period: Noon)
(Time range: PM11:00–12:00)
(Location: Dining room)
(Area ID: 21885)
(Cell ID: 10412)
(Profile: Meeting)
(Movement: Not moving)
(Battery level: High(50%–80%))
(Battery level: Full(>80%))
(Application: Calendar)
(Application: Clock)

7.3.3 Case study A

We manually analyze the mined contexts for intuitively understanding how the learning result of topic models outperforms that of K-means. An example is shown as follows. First, we contact one volunteer and know he has a typical personalized context that "*having lunch in the dining room on week day expect for Thursday*". Thursday is an exception because the volunteer has lessons in another campus on Thursday. We find all of LDAC, MUC, and K-means mine a context corresponding to it. For simplicity, we denote the context learned by K-means as c_a, denote the context learned by MUC as c_b, and denote the context learned by LDAC as c_c.

Table 5 shows c_a which is in the form of a group of contextual feature-value pairs. The location ID has been translated to meaningful locations to ease understanding. The most of the contextual feature-value pairs of c_a are reasonable, such as (*Day name: Monday*), (*Day period: Noon*), (*Location: Dining room*). But the volunteer points out that it also contains two noisy contextual feature-value pairs, namely (*Application: Calendar*) and (*Application: Clock*), and misses some more relevant contextual feature-value pairs such as (*Day name: Wednesday*) and (*Day name: Friday*). Thus, it is labeled with "Good". The reason is that K-means cannot cluster semantic similar context sessions [9]. In contrast, topic model-based approaches have a theoretical generative process to interpret the data and thus do not have such a shortage. Table 6 and Table 7 lists all contextual feature-value pairs in c_b and c_c, respectively. From the tables, we can see that all listed contextual feature-value pairs are sensible to represent the user context. As expected, they are labeled with "Perfect".

7.3.4 Case study B

We also manually analyze some mined contexts for intuitively understanding how LDAC's learning result outperforms that of MUC. An example is shown as follows. First, we contact one volunteer and know he has a typical personalized context that *he usually plays basketball in weekends' afternoon (PM14:00-17:00)*. Then, we manually check the learned contexts of MUC and LDAC and find that both of them discover a group of contextual feature-value pairs corresponding to that context. For simplicity, we denote the context learned by MUC as c'_a and denote the context learned by LDAC as c'_b.

Table 6 Context c_b learned by MUC

(Is a holiday?: No)
(Day name: Monday)
(Day name: Tuesday)
(Day name: Wednesday)
(Day name: Friday)
(Day period: Noon)
(Time range: PM11:00–12:00)
(Location: Dining room)
(Area ID: 21885)
(Cell ID: 10412)
(Profile: Meeting)
(Movement: Not moving)
(Battery level: High(50%–80%))
(Battery level: Full(>80%))

Table 7 Context c_c learned by LDAC

(Is a holiday?: No)
(Day name: Monday)
(Day name: Tuesday)
(Day name: Wednesday)
(Day name: Friday)
(Day period: Noon)
(Time range: PM11:00–12:00)
(Location: Dining room)
(Area ID: 21885)
(Cell ID: 10412)
(Profile: Meeting)
(Movement: Not moving)
(Battery level: High(50%–80%))
(Battery level: Full(>80%))

Table 8 shows c'_a which is in the form of a group of contextual feature-value pairs. The location ID has been translated to meaningful locations to ease understanding. The most of the contextual feature-value pairs of c'_a are reasonable, such as (*Day name: Saturday*), (*Day period: Afternoon*), and (*Location: Basketball area*). But it also contains two noisy contextual feature-value pairs, namely, (*Time range: PM13:00-14:00*) and (*Time range: PM17:00-18:00*); in these time range, the volunteers did not stay in the Basketball area, and it also misses some more relevant contextual feature-value pairs such as (*Time range: PM14:00-15:00*) and (*Time range: PM15:00-16:00*). Thus, it is labeled with "Good". Table 9 lists all contextual feature-value pairs in c'_b. From this table, we can see that all listed contextual feature-value pairs are sensible to represent the user context. As expected, c'_b is labeled with "Perfect". Actually, the performance of MUC directly depends on the stage of context segmentation. Since different users may have different activity patterns, it is hard to set a universal optimal ξ and the segmented context sessions may not accurately capture the context transition points. Some different context sessions may be merged, or one context session may be split, both of which will cause the irrelevant data in the result of MUC. In contrast, LDAC has a more

Table 8 Context c'_a learned by MUC

(Is a holiday?: Yes)
(Day name: Saturday)
(Day period: Afternoon)
(Time range: PM13:00–14:00)
(Time range: PM16:00-17:00)
(Time range: PM17:00–18:00)
(Location: Basketball area)
(Area ID: 21761)
(Cell ID: 10066)
(Profile: Outdoor)
(Movement: Not moving)
(Battery level: High(50%–80%))
(Battery level: Full(>80%))
(Inactive time: Middle(5–30 minutes))

Table 9 Context c'_b learned by LDAC

(Is a holiday?: Yes)
(Day name: Saturday)
(Day name: Sunday)
(Day period: Afternoon)
(Time range: PM14:00–15:00)
(Time range: PM15:00–16:00)
(Time range: PM16:00–17:00)
(Location: Basketball area)
(Area ID: 21761)
(Cell ID: 10066)
(Profile: Outdoor)
(Movement: Not moving)
(Battery level: Full(>80%))
(Inactive time: Middle(5–30 minutes))

reasonable assumption that each context session may include multiple contexts, and thus can avoid the problem.

8 Conclusion and future work

In this paper, we proposed an unsupervised approach to mobile context modeling, which is a fundamental research problem toward leveraging the rich contextual information of mobile users to support personalized customer experiences. Specifically, we first extracted context sessions from the raw context data of mobile users and then exploited two methods for learning personalized contexts from context sessions. In the first method, we cluster context sessions and extract the frequent contextual feature-value pairs from context session clusters as contexts. In the second method, we extended topic models to learn personal mobile contexts from the context sessions. Two topic models have been extended and exploited for mobile context modeling, namely MU and LDA. Experiments results on real-world context data show that the topic model-based method outperforms the clustering-based method in terms of the effectiveness of mining personalized contexts but less efficient than the latter in terms of the computational cost. Moreover, among the two topic models, the LDA-based model is more effective than that extended from MUC for mobile context modeling but less efficient than the latter in terms of the computational cost.

As for future work, it is desirable if we can incorporate some domain knowledge of common contexts, such as "waiting a bus" or "having a dinner", with unsupervised approaches for mobile context modeling. Such a semi-supervised approach may improve the learning performances of common contexts while keeping the flexibility of supervised approaches for learning personalized contexts.

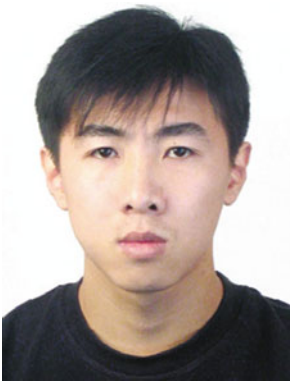
Acknowledgments The work described in this paper was supported by grants from Natural Science Foundation of China (Grant No. 6107311060775037), the Key Program of National Natural Science Foundation of China (Grant No. 60933013), and Research Fund for the Doctoral Program of Higher Education of China (20093402110017).

References

1. Abowd GD, Atkeson CG, Hong J et al (1997) Cyberguide: a mobile context-aware tour guide. Wirel Netw 3(5):421–433
2. Aggarwal C (2011) A segment-based framework for modeling and mining data streams. Knowl Inf Syst. Online first, 2010-11-22
3. Anagnostopoulos C, Tsounis A, Hadjiefthymiades S (2007) Context awareness in mobile computing environments. Wirel Pers Commun 42(3):445–464
4. Azzopardi L, Girolami M, Van Risjbergen K (2003) Investigating the relationship between language model perplexity and IR precision-recall measures. In: Proceedings of the 26th international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR'03), pp 369–370
5. Becchetti L, Colesanti U, Marchetti-Spaccamela A et al (2010) Recommending items in pervasive scenarios: models and experimental analysis. Knowl Inf Syst. Online first, 2010-09-08
6. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Research 3:993–1022
7. Eagle N, Pentland A (2009) Eigenbehaviors: identifying structure in routine. Behav Ecol Sociobiol 63:1057–1066
8. Ester M et al (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD'96, pp 226–231
9. Farahat A, Kamel M (2010) Statistical semantics for enhancing document clustering. Knowl Inf Syst. Online first, 2010-12-15
10. Han J, Kamber M (2006) Data mining: concepts and techniques, 2nd edn, chap 7. Cluster analysis. Morgan Kaufmann
11. Hearst MA (1997) Texttiling: Segmenting text into multi-paragraph subtopic passages. Computational Linguistics, pp 33–64
12. Heinrich G (2008) Parameter estimation for text analysis. Technical report, University of Leipzig, 2008
13. Hermes L, Buhmann JM (2003) A minimum entropy approach to adaptive image polygonization. IEEE Trans Image Process 12:1243–1258
14. Himberg J, Korpiaho K, Mannila H et al (2001) Time series segmentation for context recognition in mobile devices. In: Proceedings of the 2001 IEEE international conference on data mining (ICDM'01), pp 203–210
15. Hofmann T (1999) Probabilistic latent semantic analysis. In: Proceedings of the 1999 uncertainty in artificial intelligence (UAI'99), pp 289–296
16. Lemlouma T, Layaïda N (2004) Context-aware adaptation for mobile devices. In: Proceedings of IEEE international conference on mobile data management, pp 106–111
17. Liao L, Patterson DJ, Fox D et al (2005) Building personal maps from gps data. In: Proceedings of IJCAI workshop on modeling others from observation
18. Mohri M, Moreno P, Weinstein E (2010) Discriminative topic segmentation of text and speech. In: Proceedings of the 13th international conference on artificial inteligence and statistics (AISTATAS'10)
19. Nigam K, McCallum A, Thrun S (2000) Text classification from labeled and unlabeled documents using EM. Mach Learn 39:103–104

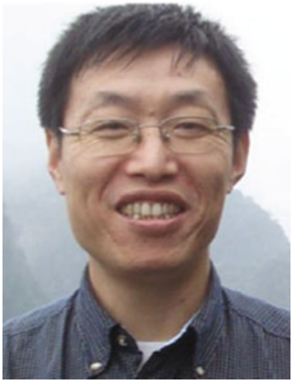
20. Otzturk P, Aamodt A (1997) Towards a model of context for case-based diagnostic problem solving. In: Context-97: proceedings of the interdisciplinary conference on modeling and using context, pp 198–208
21. Resnik P, Hardisty E (2010) Gibbs sampling for the uninitiated. Technical report, University of Maryland
22. Schilit B, Adams N, Want R (1994) Context-aware computing applications. In: Proceedings of the workshop on mobile computing systems and applications, pp 85–90, IEEE Computer Society
23. Terzi E, Tsapara P (2006) Efficient algorithms for sequence segmentation. In: Proceedings of the 2006 SIAM conference on data mining (SDM'06)
24. Van Setten M, Pokraev S, Koolwaaij J (2004) Context-aware recommendations in the mobile tourist application compass. Adaptive hypermedia and adaptive web-based systems, pp 235–244
25. Xu X Yuruk N, Feng Z et al (2007) Scan: a structural clustering algorithm for networks. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'07), pp 824–833, ACM
26. Zhang T et al (1996) Birch: an efficient data clustering method for very large databases. In: SIGMOD'96, pp 103–114
27. Zheng Y, Liu L, Wang L et al (2008) Learning transportation mode from raw GPS data for geographic applications on the web. In: Proceeding of the 17th international conference on World Wide Web Conference (WWW'08), pp 247–256

Author Biographies



Tengfei Bao received a B.E. degree from University of Science and Technology of China in 2007. He is currently a Ph.D. student at the school of Computer Science, University of Science and Technology of China. This work is carried out when he was an intern of Nokia Research Center, Beijing laboratory. His research interests include mobile context modeling and mobile user geo-trajectory mining.

Huanhuan Cao is currently a senior researcher of Nokia Research Center, Beijing lab. He received a B.E. degree and a Ph.D. degree from University of Science and Technology of China, China, in 2005 and 2009, respectively. His major research interests include mobile context mining and mobile user behavior analysis. Dr. Cao used to be a long term intern of Microsoft Research Asia from Oct. 2007 to Apr. 2009. When working in MSRA, he did a series of studies of leveraging large scale search logs for supporting context-aware search. Due to his academic achievement in this field, he won the Microsoft Fellow and Chinese Academic Science President Award.



Enhong Chen is a professor and a vice dean at the School of Computer Science and Technology, University of Science and Technology of China, IEEE Senior Member. He received his doctor degree in USTC in 1996. He was selected for Program for New Century Excellent Talents in University (supported by Chinese Ministry of Education) in 2005. He is the vice-chair of Multimedia Computing and Communication Ministry of Education-Microsoft key Laboratory. He serves as a councilor of Machine Learning Society of Chinese Association for Artificial Intelligence, and a councilor of Artificial Intelligence and Pattern Recognition Society of the Database Society of Chinese Computer Federation. Prof. Chen has been actively involved in the research community by serving as a PC member for more than 30 conferences, such as ICTAI 2006, ICTAI 2007, AIRS2009, AIRS2010, and KDD2010. One of his papers received the Best Application Paper Award in KDD2008. His research interests include semantic web, machine learning and data mining, web information processing, and constraint satisfaction problem.

Jilei Tian received his B.S., M.S. degrees in Biomedical Engineering from Xi'an Jiaotong University, China, and Ph.D. degree in Computer Science from University of Kuopio, Finland, in 1985, 1988, and 1997, respectively. He has been with Nokia Research Center as a senior and principal researcher since 1997, and recently as research leader, primarily in the area of spoken language processing, context data intelligence and economics. He also manages European project (FP7) on personalized speech-to-speech translation in Nokia. Prior to joining Nokia Research Center, he was assistant lecturer and lecturer in Institute of Information Science in Beijing Jiaotong University during 1988-1993. From 1993 to 1994, he worked at Beijing office of Mainlife Co. (HongKong) responsible for developing Chinese market of scientific instruments. He has authored or coauthored over 70 refereed publications including book chapter, journal, and conference papers and holding 50 granted and pending patents. He has served as member of technical committee and technical review committee for conferences and workshops.

Hui Xiong is currently an Associate Professor in the Management Science and Information Systems department at Rutgers University. He received the B.E. degree from the University of Science and Technology of China, China, the M.S. degree from the National University of Singapore, Singapore, and the Ph.D. degree from the University of Minnesota, USA. His general area of research is data and knowledge engineering, with a focus on developing effective and efficient data analysis techniques for emerging data intensive applications. He has published over 80 technical papers in peer-reviewed journals and conference proceedings. He is a coeditor of Clustering and Information Retrieval (Kluwer Academic Publishers, 2003) and a coeditor in chief of Encyclopedia of GIS (Springer, 2008). He is an Associate Editor of the Knowledge and Information Systems journal and has served regularly in the organization committees and the program committees of a number of international conferences and workshops. He is a senior member of the IEEE, and a member of the ACM.