Hierarchical Latent Context Representation for CARS

Moshe Unger, and Alexander Tuzhilin, Stern School of Business, NYU

Abstract—In this paper, we propose a hierarchical representation of latent contextual information that captures contextual situations in which users are recommended particular items. We also introduce an algorithm that converts unstructured latent contextual information into structured hierarchical representations. In addition, we present two general context-aware recommendation algorithms that extend collaborative filtering (CF) approaches and utilize structured and unstructured latent contextual information. In particular, the first algorithm utilizes structured latent contexts and the second one combines the structured and the unstructured latent contextual representations. By using latent contextual information in a recommendation model, we capture and represent both the structure of the latent context in the form of a hierarchy and the values of contextual variables in the form of an unstructured vector. We tested the two proposed methods with two CF-based methods on several context-rich datasets under different experimental settings. We show that using hierarchical latent contextual representations leads to significantly better recommendations than the baselines for the datasets having high- and medium-dimensional contexts. Although this is not the case for the low-dimensional contextual data, the hybrid approach, combining structured and unstructured latent contextual information, significantly outperforms other baselines across all the experimental settings and dimensions of contextual data.

Index Terms—context-aware recommender system, context, matrix factorization, hierarchical clustering.

1 Introduction

Recommender systems (RSs) have become one of the major applications that aim to tailor items to the user's preferences. Traditional recommendation algorithms capture users' interests and their interactions with items without taking into account contextual information, such as time and location. However, user interests may change depending on the context [1]. In real-life applications, there is plenty of information regarding user's circumstances and surroundings (e.g., the activity of the user, time, location, weather, etc.). Such contextual information can be high-dimensional and is gathered from multiple sources, such as web pages, mobile devices, and more. RSs taking context information into account are called context-aware recommender systems (CARSs) [1].

Most of the traditional CARSs have used pre-defined explicit contextual information for the recommendation purposes [2], [3], such as the circumstances of the information being collected, e.g., weather conditions (sunny, cloudy, raining, etc.), time conditions (weekday, weekend, etc.) or locations where recommendations are provided (e.g., midtown Manhattan). While explicitly specified contexts usually are of small dimensionality encompassing only few contextual variables handpicked for a specific application domain, they may not represent the most effective and all-encompassing set of contextual features for the recommendation application. In order to address these limitations, recent studies [4], [5], [6], [7], [8] proposed to use *latent contexts* in CARSs that are usually modeled using embeddings, which constitute implicitly defined latent vectors obtained using various dimensionality reduction methods [9], [10]. It has been recently shown that these latent approaches to CARSs significantly improve performance of RSs.

However, the previously proposed approaches modeled latent contextual information as *vectors* of high dimensionality, and thus, ignored certain structure of latent contextual variables. In particular, these methods, while trying to reduce dimensionality of the contextual space, do not take into account the structure of latent contextual variables and the semantically meaningful interrelationships among them [11], [12]. Note that explicit contextual features usually have a hierarchical structure, such as time that can be divided into days, weeks, months and years, while traditional latent context-aware approaches do not consider hierarchies. In this paper we argue that it is also important to model latent contextual variables using hierarchical representations and that these representations can automatically be inferred by the system.

We propose a new structured representation of latent contexts that is organized in a hierarchical manner and includes groups of similar latent contexts, called *contextual situations*. For example, if a latent contextual vector represents explicit contextual factors of ″noon,″ ″loud,″ ″not moving″ and ″location = (Warren Weaver Hall, NYU),″ this set of contexts collectively defines the contextual situation of a student attending a class lecture at the Warren Weaver Hall of NYU. Furthermore, these contextual situations can be organized into hierarchical structures by aggregating unstructured latent contextual vectors into high-level contextual representations. For instance, more granular contextual situations, such as ″attending a class lecture at NYU″ and ″eating in the cafeteria at NYU,″ can be aggregated to a higher-level contextual situation of ″being located at the university″ that is less granular than the two previous contextual situations.

By modeling latent contexts as structured hierarchies obtained from unstructured latent vectors, we can enhance recommendation algorithms with new types of contextual situations that improve recommendations. In particular, we propose a generic context-aware algorithm that extends collaborative filtering (CF) methods with hierarchical latent contexts. We demonstrate the importance of modeling hierarchical latent contexts in the proposed contextual recommendation approach by showing that it improves recommendation performance in multiple dimensional contextual spaces. We also present a hybrid model that combines both unstructured *and* structured latent contexts when recommending items to the users. We compare our recommendation algorithms with other state-of-the-art context-aware algorithms on six real world context-aware datasets and demonstrate that our approaches outperform the baselines across different experimental settings.

Modeling latent contexts as structured hierarchies provides several important advantages over modeling contexts as unstructured latent vectors. First, as is shown in the paper, this approach provides significant recommendation performance improvements - up to 13.8% of RMSE in some situations, especially in the mid- to high-dimensional cases. Second, it captures latent contextual information better than the unstructured vector-based approach, since it identifies the *patterns* of latent contextual information, i.e., contextual situations, that are useful for the recommendation purposes and for the conceptual representation and understanding of latent contexts (as is shown in Section 5). Contextual situations are important beyond their applicability in RSs and can be used by marketers to understand complex contexts of customer actions and by data scientists in general for developing better predictive models. Finally, structured latent contextual information is more *compact* than the unstructured latent vectors, which helps to overcome the major sparsity problem of CARSs [1], [13].

The main contributions of this paper are as follows. First, we propose a novel representation of implicit context that models latent contextual information in a hierarchical manner. This representation is obtained from unstructured latent contexts, is capable of modeling the recurrent patterns of the latent contextual space, and captures complex contextual situations of the user at different granularity levels. Second, we demonstrate the impact that dimensionality of contextual information has on the recommendation accuracy. In particular, by extending existing CF-based models with hierarchical latent contexts, we can obtain significantly better recommendation accuracy for the rating prediction task when the contextual space is medium- or high-dimensional, while in a low-dimensional contextual space the improvements of hierarchical latent contexts over the unstructured latent contexts are smaller. Third, we enhance CARSs by proposing a hybrid latent context-aware method that incorporates both the structured and unstructured latent contexts in a complementary manner. Finally, we perform extensive offline experiments to evaluate the proposed models on six different datasets containing various contextual dimensions. The experiments show that our hybrid model, utilizing both unstructured and structured contextual information, outperforms traditional CARSs.

The rest of this paper is organized as follows: Section 2 describes related work, and Section 3 describes the contextual recommendation model that includes structured and unstructured latent contexts. Section 4 presents the datasets, the baselines and the evaluation protocols, and Section 5 presents a case study of analyzing hierarchical latent contexts. Finally, in Section 6 we discuss the results and in Section 7 we discuss the limitations of our method and plans for future work.

2 Related Work

2.1 CARSs (Context-Aware Recommender Systems)

The area of context-aware recommender systems (CARSs) [1], [13] has drawn much research attention in recent years due to the emergence and penetration of smart mobile devices that utilize sensors to collect available data about users [14]. CARS deals with modeling and predicting user preferences by incorporating available contextual information into the recommendation process. In recent years, researchers have shown that the integration of contextual information (e.g., time, place and weather) in recommender systems (RSs) improves recommendation accuracy [13].

Existing CARSs methods can be grouped into three main paradigms [1]: contextual pre-filtering, contextual post-filtering and contextual modeling. While the pre- and post-filtering approaches are used in traditional context-unaware recommendation algorithms to select the relevant set of rating data before (or after) computing predictions, their main limitation comes from the difficulty to obtain ratings in all of the possible explicit contextual situations, which makes it hard to build a robust and contextualized rating prediction model. In the contextual modeling approach, contextual information is directly incorporated into the prediction model as part of the rating model. Several CARSs based on contextual modeling methods have been proposed so far, such as the extension of Matrix Factorization (MF) [15] and Factorization Machines (FM) [16] with different influences of contextual information on items and users [3], [17]. These methods utilize explicit contextual information, which refers to the user's current situation from a reduced set of known labeled contexts (e.g., at work, running, etc.) that is selected manually by domain experts.

In contrast to the described explicit approaches of CARS, several studies [7], [8] proposed to generate an effective and reduced set of contextual features for the recommendation process, referred as "latent contexts", that model the implicit context of the user and can significantly improve recommendation accuracy. Recently, deep learning has revolutionized recommendation architectures [18], [19], [20]. In particular, several context-aware deep learning models have been suggested: [4] proposed utilizing convolutional neural networks (CNNs) for enhancing FM and models high-order interactions between contextual variables, and [21] aims to learn latent contextual features that reflect user's preferences over all candidate items and considers explicit contextual features. [22] proposed a context-aware session-based recommendation utilizing conditional recurrent neural networks (RNNs) that injects contextual information into input and output layers and modifies the behavior of RNNs. [5] extracted unstructured latent contexts from rich contextual factors such as images and texts for event prediction and [23] extracted latent content information from documents using convolutional neural network. While these works strengthen the hypothesis that explicit and latent contextual information improves recommendation accuracy, they modeled contextual information as *vectors* of high dimensionality and ignored certain structure of latent contextual variables.

In recent years, some works have considered structures of latent information in RSs [24], [25], [26]. Although these works integrate various structures, such as trees, with latent factor models, they do not incorporate contextual information into recommender systems. [8] proposed a context-aware approach that extends tensor factorization by using encoded contextual features extracted from a regression tree. While this work considered hierarchical contextual features, their method has several limitations: (1) since tensor factorization can handle a small number of contextual features, they represented contextual features by only one cluster (i.e., the tree leaf), while more complex contextual patterns can be revealed by a richer structured representation. (2) Clustering was applied on all available explicit contextual variables with low-dimensional datasets (up to 12 contextual features). However, in environments with rich and high-dimensional context space, clustering becomes inefficient and challenging [27] and context dimension must be reduced. In addition, although clustering has the potential to group similar contextual features at some level, it does not reveal the relations among the features and limits the analysis to a fixed number of learned clusters. (3) A regression tree was trained with a fixed number of tree depth, which can be hard to define in advance.

We suggest leveraging the advantages of each the studies above [7], [8], [24] and propose to model latent contextual variables using hierarchical representations in order to automatically capture complex contextual situations at different granularity levels. We also present a novel context-aware recommendation model that extends traditional CF-based models with both structured (hierarchical) and unstructured latent contextual information. We evaluate our method on six context-aware datasets and discuss how the dimensionality of the contextual space can affect the recommendation accuracy. A detailed model and algorithms are described in Section 3.

2.2 Hierarchical Context Information

In most real-life context-aware applications that use contextual information and adapt their functionality to the users' immediate context, there is plenty of contextual information that describe the current situation of a user in terms of location, activity and surroundings. For example, mobile devices contain a large set of embedded sensors that can be used for context inference and are utilized to identify a wide range of user-related contexts or activities. [28] examines activity recognition from mobile sensed information referring to physical movements (e.g., walking, driving) and physical activities (e.g. eating, studying). Other studies refer to environmental characteristics (e.g. cold, warm) and emotional conditions of the user (e.g. nervous, happy, excited).

The contextual information that is used to determine the situation of the user can be organized as a structured representation of trees [29], [30]. [31] modeled contextual information in a hierarchical structure and used a specialization taxonomy to build the tree. They handled only explicit contextual features with categorical values, which is not practice in real-life solutions. These limitations with explicit contexts may cause serious problems in several practical applications, such as smart health and well-being, mobile sensing and internet-of-things (IoT) [28], where the contextual feature space is high-dimensional, complex and dynamic. For example, by exploiting numerous sensors derived from smartphones, such as accelerometer, magnetic field, GPS and light, high-dimensional contextual information (up to 600 contextual variables) can be collected automatically to infer users' behaviors and contexts [14]; the accelerometer sensor can be used to infer the activity of the user (e.g., walking or sitting), while other features from the GPS sensor can be used to infer her location (e.g., at work, at home, etc.).

Identifying all the explicit contexts that are relevant for the service can be very challenging, especially in big data environments, and therefore automatic latent modeling of contextual information can be useful to capture the richness of the contextual information [5] and can *implicitly* identify various types of contexts. Previously latent context-aware approaches [6], [12], [32] modeled latent contextual information as unstructured vectors of high dimensionality, and ignored certain structure of latent contextual variables. The key difference between our work and methods mentioned above is that we model latent contexts as structured hierarchies in order to better reflect their structure and identify the patterns of latent contextual information, i.e., contextual situations, that are useful for the recommendation purposes and for the conceptual representation and understanding of latent contexts. We use all available contextual features (e.g., weather, sound, light, location) in order to extract a latent and compressed representation of context, and then we build an efficient tree out of the latent context instances. In this way, we transform the latent contextual space to a richer structured contextual representation at different granularity levels. This structured representation of latent contexts reveals implicit contextual situations that differ from one another, called "*hierarchical latent contexts*".

3 METHOD

In this section, we present a latent context-aware recommendation model utilizing a new representation of structured and unstructured latent contextual information. Specifically, we suggest building a hierarchical structure for representing latent contextual information that captures complex contextual situations at different granularity levels. This model is called *latent contextual situations hierarchy* and organizes compressed latent contextual vectors into groups of implicit contextual situations. By modeling latent contextual vectors in a structured manner from a non-hierarchical latent contextual information, we can enrich the recommendation algorithm with new types of context patterns that are compact and better reflect the user's context. In addition, when using latent contextual information in a recommendation model, we would like to capture and represent both the structure of the latent context in the form of a hierarchy and the values of contextual variables in the form of a latent vector. Then we use this structured and unstructured latent contextual representation in developing our recommendation method

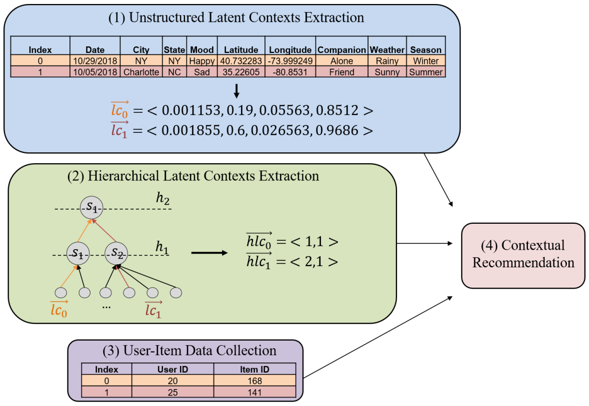


Fig. 1. Method overview.

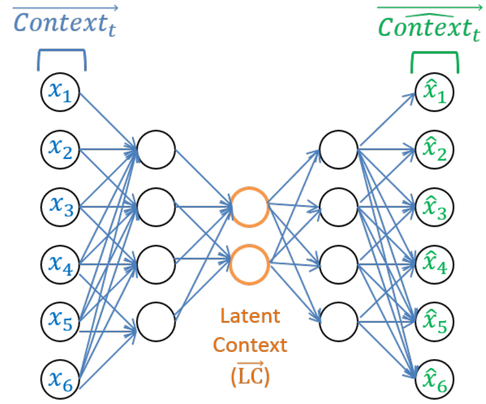


Fig. 2. Example of an auto-encoder network structure for extracting unstructured latent contexts.

that is presented in Figure 1 and consisting of the following steps:

(1) Unstructured Latent Contexts Extraction step is responsible for extracting latent contextual vectors from the collected contextual feature space. We extract a reduced representation of latent contexts from multiple contextual dimensions, such as mobile sensors, weather and location. This step includes three parts: (1) collecting raw contextual data (e.g., from mobile devices we may collect data from the accelerometer and Wi-Fi, and light and microphone information); (2) extracting a set of contextual features from the collected raw data (e.g., calculating statistics such as minimum, maximum, and average of the collected accelerometer records), and (3) reducing the high-dimensional contextual space to a compressed latent contextual space using classical unsupervised learning techniques (e.g., auto-encoding and PCA). As a result of this step, we obtain an unstructured contextual latent vector $\overrightarrow{lc_i}$ that reflects the current user context in an implicit manner, as shown in Step 1 of Figure 1. Note that dimensionality of the latent contextual space is compressed to a limited number of latent values.

(2) Hierarchical Latent Contexts Extraction step extracts a new structured context representation from the latent context vectors obtained in Step 1. This structured context is organized in hierarchical manner and includes groups of similar latent contexts, called *contextual situations*, that are organized at different granularity levels. The input for this step is the latent contextual vector $\overrightarrow{lc_i}$, which is defined by a limited set of numerical values. The output is a hierarchical latent context vector $\overrightarrow{hlc_i}$, a set of implicit contextual situations, obtained from the path of the latent context vector $\overrightarrow{lc_i}$ in the constructed tree, as shown in Step 2 of Figure 1. This structured representation is much more compact than the latent contextual vector and reveals the patterns among the latent values.

(3) User-Item Data Collection step obtains information on user preferences for the items the system wishes to recommend, typically in the form of ratings, that are subsequently used by the collaborative filtering systems [15], [24], as shown in Step 3 of Figure 1.

(4) Contextual Recommendation step combines the outputs of all of the previous steps and constructs the complete rating model capturing user-item interactions, unstructured and hierarchical latent contexts.

In the rest of this section we explain each of these steps in greater detail.

3.1 Unstructured Latent Contexts Extraction

Contextual information can be described by various types of environmental features, such as time, location, weather and sensor data [1]. It has been recently shown that reducing and automatically selecting the optimal set of contextual factors in an implicit manner, called "latent contexts", can significantly improve performance of CARSs [5], [6], [7]. In order to reduce the dimensionality and noise of the feature space and reveal relationships between the contextual features, we follow [7] and extract unstructured latent contexts with two types of unsupervised learning techniques: (1) auto-encoding [10] and (2) principal component analysis (PCA) [9]. These unstructured latent contextual variables are numeric attributes representing the current context of the user in an implicit manner. In the following sections, we will briefly present the extraction of unstructured latent contexts using the two techniques.

3.1.1 Unstructured Latent Contexts Extraction Using Auto-Encoder

Since context often consists of environmental and sensorial data (such as location, accelerometer data, light, etc.), many of its variables being correlated with other variables. Hence, we use an auto-encoder (AE) [10] to discover non-linear correlations between different features and extract a latent representation out of these correlations. This latent representation is generated from the compressed layer of a trained AE (i.e., $\overrightarrow{LC}$ layer in Figure 2). Note that the auto-encoding process is performed by an unsupervised learning algorithm that applies backpropagation that is aimed at setting the value of the targets to be equal to the value of the inputs [19]. In the training phase, the encoder tries to learn the function $h_{W,b}(\overrightarrow{Context_t}) \simeq \overrightarrow{\hat{Context_t}}$, where W and b are the weights of the network's edges. By putting constraints on the neural network, we can discover an interesting data structure, which is composed of limited numbers of hidden units in each layer forcing the network to learn a compressed latent representation of the contextual input.

3.1.2 Unstructured Latent Contexts Extraction Using Principal Component Analysis (PCA)

The second dimensionality reduction method that we apply utilizes Principal Component Analysis (PCA) [9]. It extracts latent contextual attributes by performing an orthogonal transformation to convert the set of original contextual variables, possibly having collinear dependencies, into a set of new uncorrelated latent features (principal components), each new feature being a linear combination of the original features. PCA entails (a) computing the covariance matrix of the original features (after a normalization process), (b) calculating the eigenvectors and eigenvalues of the covariance matrix, (c) ordering the eigenvectors according to their matching eigenvalues, and (d) selecting eigenvectors corresponding to top-k largest eigenvalues. We decided to use PCA for extracting latent contextual information because (a) this is a popular rigorous dimensionality reduction method and (b) it is linear- in contrast to the auto-encoder method described in Section 3.1.1.

3.2 Hierarchical Latent Contexts Extraction

The goal of this step is to model latent contextual information in a hierarchical manner in order to learn the most *recurrent* contextual situations from unstructured latent context vectors. For instance, if a latent context vector represents the values of contextual factors such as "noon," "loud," "not moving" and "location = (Warren Weaver Hall, NYU)," this set of contexts collectively represents a recurrent situation of attending a class lecture at NYU. Unlike unstructured latent contexts, which represent a reduced set of non-hierarchical contextual values reflecting the current context of the user, hierarchical latent context is a set of *contextual situations* specified at different granularity levels. For instance, more granular contextual situations such as "attending a class lecture at NYU" and "eating in the cafeteria at NYU" can be aggregated to a higher-level contextual situation of "being located at the university" that is less granular than the two previous contextual situations.

The process of constructing a hierarchical model is done by grouping the set of all unstructured latent contextual vectors into a finite set of clusters, each cluster representing a particular implicit contextual situation. We apply agglomerative hierarchical clustering (AHC) [33] to the latent contextual vectors in order to automatically estimate the number of contextual situations S that are represented as clusters. Then we apply the k-means algorithm to group similar latent contextual vectors into a contextual situation. The hierarchical clustering approach would produce a hierarchical tree structure, such as the one presented in Figure 3. As Figure 3 shows, for each latent contextual vector $\overrightarrow{lc_j}$ in the leaf of the tree, we find node $s_{h_t i}$ at the hierarchy level h_t that is an ancestor of leaf node $\overrightarrow{lc_j}$. This node $s_{h_t i}$ represents a similar contextual situation for the latent contextual vector $\overrightarrow{lc_j}$ at level h_t of the tree hierarchy. The final hierarchical latent representation is the path of $\overrightarrow{lc_j}$ from the leaf to the top of the tree, which results in a set of contextual situations $s_{h_t i}$ at each hierarchy level $h_t \in H$.

The levels of the tree hierarchy H are selected based on how similar (or dissimilar) the situations (set of latent contextual vectors) are. The similarity is computed

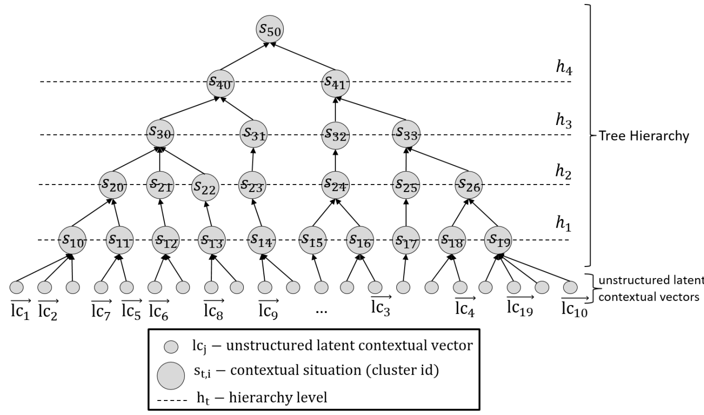


Fig. 3. Latent contextual situations hierarchy structure for obtaining hierarchical latent contexts.

by the Ward's minimum variance method [33]. Situations that share the same ancestor in the same hierarchy level h_t are similar to each other and share the same contextual patterns. In order to produce a hierarchical latent contextual representation from the learned tree, we extract the hierarchical latent contextual vector $\overrightarrow{hlc} = [s_{1i}, s_{2j}, ..., s_{|H|t}]$ for each latent contextual vector $\overrightarrow{lc_j}$, each contextual situation s_t reflects the similar situation for $\overrightarrow{lc_j}$ at level h_t of the tree hierarchy. For example, the extracted hierarchical latent contextual vector for $\overrightarrow{lc_{19}}$, based on the tree in Figure 3, is $\overrightarrow{hlc_{19}} = [s_{19}, s_{26}, s_{33}, s_{41}]$, corresponding to the hierarchy levels h_1, h_2, h_3 and h_4.

Algorithm 1 describes the process of building a hierarchical tree and extracting hierarchical latent contexts. The input for the algorithm is the training set $LC = [\overrightarrow{lc_1}, \overrightarrow{lc_2}, ..., \overrightarrow{lc_n}]$, where each latent context instance $\overrightarrow{lc_j}$ is a compressed r-dimensional vector, containing r numerical features that were extracted from the contextual features based on auto-encoding or PCA (as described in Section 3.1). The second input is the stopping threshold α for the AHC algorithm that is used to estimate the number of contextual situations (clusters) S. The output of the algorithm is the set of n hierarchical latent contextual vectors $res = [\overrightarrow{hlc_1}, \overrightarrow{hlc_2}, ..., \overrightarrow{hlc_n}]$ for each of the corresponding latent contextual vector $\overrightarrow{lc_i}$ in LC. $\overrightarrow{hlc_i}$ is a vector that composed of $|H|$ contextual situations for each level $h_t \in H$ of the tree hierarchy. H is determined automatically by the structure of the tree and reflects the hierarchy levels of the tree.

The algorithm starts with applying AHC with latent contextual vectors in order to estimate the number of contextual situations S (line 2). The construction of the hierarchical tree by applying AHC leads to a complexity of $O(n^2)$ for n latent contextual vectors. Then k-means clustering is applied in order to improve the initial clustering obtained with AHC (see lines 3 to 15). The k-means is an iterative relocation algorithm until squared error achieves its minimal value [33]. In lines 4,12 the function $NewMean$ calculates the mean of all the latent contextual vectors in the contextual situation s_i. The time complexity of the k-means is $O(t*s*n*r)$, for the n (r-dimensional) latent contextual vectors, where $s = |S|$ is the number of clusters (contextual situations). In line 16, we compute the hierarchy levels of the tree by the updated contextual situations S and the set of latent contextual vectors LC. This computing is done by selecting only the hierarchy levels $h_t \in H$ in the tree that obtain more than 60% in their variance values ($h_t \geq h_{t-1} * 1.6$) that are computed by the Ward's variance method [34] ($O(H)$). Then, for each latent contextual vector, we extract the hierarchical latent contexts representation (see lines 17-20), which is computed by its contextual situation in each level of the tree hierarchy. This leads to a time complexity of $O(H*n)$ for n latent contextual vectors and H hierarchy levels. The result res (line 23) is a set of hierarchical latent contexts, each of them is $|H|$-dimensional and contains contextual situations at different granularity levels of the tree.

The overall complexity of this step is $O(n^2)$, since the construction of the hierarchical tree by applying AHC and updating the contextual situations by using k-means requires the computation of all pairwise similarities between n latent contextual vectors.

Algorithm 1 Extracting Hierarchical Latent Contexts

Input: the training set LC that contains n latent contextual vectors; the threshold α.
Output: the set of hierarchical latent contexts res.

1: $O \leftarrow \emptyset, error \leftarrow \infty$
2: $S \leftarrow AHC(LC, \alpha)$
3: **for** $s_i \in S$ **do**
4: Update its mean $mean_{s_i} = NewMean(s_i)$
5: **end for**
6: **repeat**
7: $error_{old} = error$
8: **for** latent contextual vector $\overrightarrow{lc_j} \in LC$ **do**
9: Assign $\overrightarrow{lc_j}$ to the closest contextual situation $s \in S$, such that $\forall s_i \in S$, $MATCH(lc_j, s_i) \leq MATCH(lc_j, s)$
10: **end for**
11: **for** contextual situation $s_i \in S$ **do**
12: Update $mean_{s_i} = NewMean(s_i)$
13: **end for**
14: Compute $error = \sum_{i=1}^{k} \sum_{lc \in s_i} |MATCH(lc, mean_{s_i})|^2$
15: **until** $error_{old} \leq error$
16: $H \leftarrow GetTreeHierarchy(S, LC)$
17: **for** $\overrightarrow{lc_j} \in LC$ **do**
18: **for** level of hierarchy $h_t \in H$ **do**
19: $hlc_t =$ assign $\overrightarrow{lc_j}$ to the closest contextual situation $s_{h_t} \in S$ at tree level h_t
20: **end for**
21: $res = res \cup \overrightarrow{hlc}$
22: **end for**
23: **return** res
=0

3.3 User-Item Data Collection

In this subsection, we give an overview of the rating model capturing user-item interactions, unstructured latent contextual vectors and structured (hierarchical) latent contexts. We design a general latent context-aware recommendation model, in which any CF-based model can be extended to incorporate structured contextual information. We also suggest a hybrid recommendation model that combines two types of latent contextual representations in a complementary manner: unstructured and structured latent contexts. The unstructured latent contextual representation is the same as used in [7], where the original contextual features are compressed by auto-encoding or PCA. The new recommendation model adds a structured representation of latent contexts, modeled in a hierarchical manner, in order to describe multiple contextual situations at different granularity levels. These situations describe the most recurrent contextual situations from the unstructured latent contextual vectors.

A single rating instance in our model has the following structure: $r_{u,i,l_1,...,l_k,hlc_1,hlc_2,...,hlc_h}$, where r is the actual rating score, $u \in U$ is the user ID, $i \in I$ is the item's ID, $l_1, ..., l_k = \overrightarrow{lc}$ is the set of unstructured latent contextual attributes, and $hlc_1, ..., hlc_h = \overrightarrow{hlc}$ is the set of hierarchical latent contextual attributes, where each hlc_t is the ID of a contextual situation (cluster) from a finite set of contextual situations $s_t \in S$. We normalize the contextual situation IDs to a continuous number in the range of $\{0..1\}$.

3.4 Contextual Recommendation

We present two general variations of context-aware recommendation algorithms that extend CF-based approaches and utilize structured and unstructured latent contextual information. In particular, the first is the Hierarchical Latent Context Model (HLCM) that utilizes structured latent contexts in a hierarchical manner, and the second is the Hybrid Contexts Model (Hybrid) that combines the structured and the unstructured latent contextual representations. We extend two specific CF-based models: (1) Matrix Factorization (MF) described in [15], and (2) the IHSR model described in [24], which considers implicit hierarchical structures of items and users.

A general rating prediction model is presented in Equation 1. The input for the rating prediction is a target user ID (u), a target item ID (i), the set of unstructured latent contexts ($l_1, ..., l_k$), and the set of hierarchical latent contexts ($hlc_1, ..., hlc_h$). When the recommendation model utilizes only hierarchical latent contexts (HLCM), the value of $b_{i,j}$ is 0. We add structured and unstructured latent contextual attributes and learning the set of parameters in the baseline rating prediction rule, as follows:

$$\hat{r}_{u,i,l_1,...,l_k,hlc_1,...,hlc_h} = \Theta + \sum_{j=1}^{k} b_{i,j} l_j + \sum_{s=1}^{h} b'_{i,s} hlc_s \quad (1)$$

In this model, hierarchical latent contexts hlc_s and unstructured latent contexts l_j can be added to any CF model Θ that takes into account user and item representations. The rating biases $b_{i,j}$ and $b'_{i,s}$ are used to fully reflect the impact of the latent contextual attributes of a given item i on the predicted rating. Furthermore, the unstructured and structured latent context attributes (l_j and hlc_s) are represented as continuous numbers in the range of $\{0..1\}$ in our model.

In this paper we focused on Θ being MF [15] for the following reasons: (1) due to the representation of low-dimensional hidden factors for items and users, MF is able to efficiently handle large datasets, sparseness of rating matrix and scalability problem of CF algorithm. This representation also preserves user privacy, since users and items are represented as latent factors. (2) MF easily allows incorporation of additional information, such as multiple forms of feedback, temporal dynamics, and confidence levels. Hence, generalizing our recommendation model to real-world applications is crucial for context inference and modeling for recommender systems.

In order to show the generality of our model and the importance of structured and unstructured latent contexts, we also considered IHSR model [24] as Θ, since it learns implicit hierarchies of users and items and significantly improves MF. In general, Θ can be *any* other CF model, such as memory-based approaches (user-based CF and item-based CF) [35] and model-based approaches, such as clustering-based algorithms [35]. In particular, the specific rating prediction model with Θ being MF or IHSR is presented in Equation 2:

$$\hat{r}_{u,i,l_1,\ldots,l_k,hlc_1,\ldots,hlc_h} = b_u + b_i + v_u q_i^T + \sum_{j=1}^{k} b_{i,j} l_j + \sum_{s=1}^{h} b'_{i,s} hlc_s \tag{2}$$

- b_u is the baseline estimator for user u. This parameter captures the user rating bias. For example, a user who tends to give high rating scores overall will have a high baseline.
- b_i is the baseline estimator for item i. This parameter indicates the degree of popularity or the likelihood of the given item.
- v_u and q_i are the rating score of a given user u to item i. In the extension of MF, v_u and q_i are the traditional latent factor vectors that map users and items on a set of f common latent factor space, when the multiplication of $v_u q_i^T$ is an approximation of the original and sparse rating matrix. In the extension of IHSR model, it is the rating score $X(v,q)$ from user u_v to item i_q.
- $b_{i,j} l_j$ is the rating bias of item i under the unstructured latent contextual attribute j. $b_{i,j} = 0$ for the HLCM that utilize only hierarchical latent contexts.
- $b'_{i,s} hlc_s$ is the rating bias of a given item i under the hierarchical (structured) latent contextual situation s.

We estimate the parameters of the rating model (as presented in Equation 2) by solving the following optimization problem that aims at minimizing the squared error between the predicted rating, according to the prediction rule and the actual rating:

$$\begin{aligned} & min_{v_u,q_i,b_u,b_{i,j}l_j,b_{i,s}hlc_s} \\ & = \sum_{r\in R} (r_{u,i,l_1,..,l_k,hlc_1,...,hlc_h} - b_u - b_i - v_u q_i^T - \sum_{j=1}^{k} b_{i,j} l_j \\ & - \sum_{s=1}^{h} b'_{i,s} hlc_s)^2 \\ & + \lambda(\|b_u\|^2 + \|v_u\|^2 + \|q_{u_i}\|^2 + \sum_{j=1}^{k} b_{i,j}^2 l_j + \sum_{s=1}^{h} {b'_{i,s}}^2 hlc_s) \end{aligned} \tag{3}$$

Note that the regularization term is used to minimize overfitting, and it is controlled by the parameter λ to penalize the conditions where these variables are overfitted to the observed data. To solve this optimization problem, we apply an iterative stochastic gradient descent (SGD) learning algorithm, having the following meta-parameters: R - a rating dataset with unstructured and hierarchical latent contexts, f - latent factors' size for the components v_u and q_i, k - the size of the unstructured latent contextual attributes, h - the size of the hierarchical latent contextual situations, λ - the regularization parameter, and γ - the learning step of the gradient descent algorithm.

The output of the algorithm is the learned model with values of $v_u, q_i, b_u, b_{i,j}l_j, b_{i,s}hlc_s$ for users, items, unstructured latent contexts and hierarchical latent contexts.

3.5 Method Complexity

Our method is an extension of a CF model Θ (e.g., MF) with unstructured and structured latent contextual information. In particular, we use MF [15] or IHSR [24] as the base CF model (Θ). Therefore, the time complexity of the CF model is dependent on the complexity of Θ. The integration of various contextual representations to the framework is done by representing contextual information as structured or unstructured latent context. The process of extracting latent contextual vectors is done in two steps: (1) training an auto-encoder (AE) with n contextual vectors for compressing the original contextual features from l dimensions to a r-dimensions ($O(n*l*r)$). The training of an AE is an iterative process that is performed with stochastic gradient descent through back propagation. (2) Extracting hierarchical latent contexts from the unstructured latent contextual vectors. This process includes the construction of hierarchical tree by applying agglomerative hierarchical clustering with Ward's variance method [34], which is $O(n^2)$ computational complexity for the clustering of n observations.

4 Experiments

In this section we present several experiments investigating whether our proposed latent contextual recommendation models (HLCM and Hybrid) can achieve better performance compared to other state-of-the-art recommendation methods. The experiments are performed on six different datasets described in the next section.

4.1 Datasets

We use the following six context-aware datasets in our study containing different contextual dimensions: two private datasets containing mobile data (Hearo [7] and CARS [36]), one private dataset containing music listening logs (Spotify), and three public datasets across multiple domains (LDOS-CoMoDa [37], Frappe [38] and Yelp[1]):

(1) Hearo. This high-dimensional context-aware dataset was used by Unger et al. [7] and provided by the authors. The data derived from a field experiment in which users interacted with a recommender system that provided recommendations of points of interest (POIs) obtained from the

1. https://www.yelp.com/dataset/challenge

TABLE 1
Description of Context-Aware Datasets

	Hearo Dataset	CARS Dataset	Spotify Dataset	LDOS-CoMoDa Dataset	Frappe Dataset	Yelp Dataset
num of users	77	98	7,108,598	112	957	451,341
number of items	228	1,918	4,019,103	1,232	4,082	80,796
num of ratings	7,416	38,900	1,235,653,977	2,294	96,204	3,383,536
rating scale	0-1	dislike (1), like (3), and check-in (5)	listening percent (0-1)	1-5	0-4.46	1-5
rating sparsity	57.75%	96.41%	99.99%	98.34%	97.54%	99.99%
num of contextual dimensions	16	14	5	12	3	2
num of contextual features	523	247	16	32	22	9
contextual dimensions	time, location, ringer mode, battery, activity recognition, light, accelerometer, orientation, application traffic, gravity, microphone, screen, weather, magnetic field, cell state	time, weather, ringer mode, accelerometer, orientation, location, running applications, screen log, battery, light, network traffic, gravity, microphone, magnetic field	time, location, type of device, connected resources, user intent	time, location, day type, dominantEmo, endEmo, mood, social, physical, decision, interaction, weather, season	time, location, weather	time, city

Foursquare API and received users' binary ratings about the recommendations provided. The ratings being obtained by 77 users and associated with 523 contextual features. The contextual features were extracted from multiple contextual dimensions, such as accelerometer, microphone, battery, gravity, GPS, etc. For each contextual dimension, various statistical contextual features were extracted, such as average, standard deviation, root mean square, entropy, etc.

(2) CARS. This high-dimensional context-aware dataset was used by Unger et al. [36] and provided by the authors. It contains 38,900 explicit 5-scale ratings (1-dislike, 3-like, and 5-check-in) of 1,918 Foursquare POIs, each rating being associated with 247 contextual features. The data was collected from various types of contextual dimensions that included environmental information, user activity, mobile state, and user behavioral data.

(3) LDOS-CoMoDa. This is a publicly available context-aware dataset [37] comprised of movie data collected from surveys. It contains 2,294 5-scale ratings for the movies and 32 contextual features describing the situation in which the movies were consumed, including time in a day, location, day of the week, and three emotional variables. The dataset does not include the specific timestamp of each user interaction with the system.

(4) Frappe. This implicit feedback dataset is collected from a context-aware personalized recommender of mobile apps [38]. We used 3 contextual dimensions for experimental evaluations, including time (time of the day, day of the week), location and weather. This data captures the usage frequencies of an app by each user within 2 months. We employ a log transformation on the raw frequency numbers which results in a rating scale of 0-4.46. Overall, the data contained 96,204 ratings. The dataset does not include the specific timestamp of each user interaction with the system.

(5) Yelp. This is a large-scale dataset. Since the original data was highly sparse, we retained users and items with at least 10 interactions. This results in a subset of data containing 451,341 users, 80,796 items, and 3,383,536 interactions. We use the contextual dimensions time and location and their corresponding extracted contextual features: year, month, day of the week, and city.

(6) Spotify. This is a private dataset provided by a mobile music recommendation company. The data represents one week of listening records and contains 7,108,598 users, 4,019,103 items, and 1,235,653,977 interactions. We use the following contextual features: 4 features about the user's region, 9 features about the product and platform of the user's device, day of the week, time of the day and the user interface context (home, search, library, radio, browse, other). Overall we have 16 contextual features associated with each user interaction in the system. For the rating prediction task, we predict the percent of the song listening.

For each of the datasets, we normalized the contextual features to a scale of 0 to 1 and transformed nominal features to binary features. The overall number of contextual features (after normalization) and the characteristics of the five datasets are summarized in Table 1.

4.2 Baselines

In order to test the proposed methods, we conducted a series of offline simulations with each of the datasets. We used the following four recommendation models as baselines:

- *Matrix Factorization (**MF**)* [15] - a method that characterizes both users and items by latent vectors inferred from observed ratings.
- *Implicit Hierarchical Structure (**IHSR**)* [24] - the extension of MF with implicit hierarchical structures of items and users.
- *Factorization Machines (**FM**)* [16] - a generic approach that combines the flexibility of feature engineering with factorization models. We use LibFM[2] to implement the method.
- *Convolutional Factorization Machines (**CFM**)* [4] - a deep learning approach that automatically learns feature interactions using independent embedding dimensions and implicit high-order interaction modeling. We use the explicit contextual attributes to represent user context in the model.
- *Explicit Contexts Model (**ECM**)* [17] - the extension of MF by considering different influences of explicit contextual attributes on the item bias. For explicit context features, we followed [17] and chose the time of day, time of week, and weather.
- *Latent Contexts Model (**LCM**)* [7] - the extension of MF by considering only unstructured latent context vectors that were extracted by PCA method or an auto-encoder model, as described in section 3.1.1.

And we compared them with our proposed models:

- *Hierarchical Latent Contexts Model* ($HLCM_{MF}$) - our proposed hierarchical latent model which extends MF with only hierarchical latent contextual

2. http://www.libfm.org

attributes, as described in section 3.4. $HLCM_{IHSR}$ extends the IHSR model with hierarchical latent contexts.

- *Hybrid Contexts Model* ($Hybrid_{MF}$) - our proposed model which extends MF and takes into consideration both unstructured and hierarchical latent contexts, as described in section 3.4. $Hybrid_{IHSR}$ extends the IHSR model with unstructured and hierarchical latent contexts.

4.3 Evaluation protocols

According to the principles of cross-validation, we divide each of the context-aware datasets into ten subsets, and randomly split each subset into three portions: 80% for training, 10% for validation, and 10% for testing. The validation set is used for tuning hyper-parameters for the recommendation model training, and the final performance comparison is conducted on the test set. The final results are the mean of the ten experiments. We also applied a time-based evaluation strategy with the datasets that contained the timestamp feature (i.e., Hearo, CARS, Spotify and Yelp). The splitting ratio was 80:20 for training and testing respectively, and the ratings were split in that proportion according to the ratings' timestamps. We evaluated the rating prediction accuracy with two metrics: the root mean squared error (RMSE) and the mean absolute error (MAE) [13].

For each of the datasets used in our study, we extracted latent contextual information based on the contextual features of that dataset. To do that, we trained the autoencoder (AE) network (as shown in Figure 2) containing five layers (2 input layers, 1 compressed layer of latent contexts and 2 output layers). For the CARS dataset, we trained the AE with (247, 120, 40, 120, 247) units in layers (2 input layers, 1 compressed layer and 2 output layers) respectively. For the Hearo, Spotify, LDOS-CoMoDa, Frappe and Yelp datasets, we trained the AE with (523, 260, 80, 260, 523), (16, 10, 5, 10, 16), (32, 20, 12, 20, 32), (22, 14, 8, 14, 22) and (9, 5, 3, 5, 9) units in each of the (2 input layers, 1 compressed layer and 2 output layers) layers respectively. The number of units in the compressed layer, which represents the number of unstructured latent contexts, was determined by cross-validation in a separate calibration process for each dataset. We randomly initialized the AE parameters using Gaussian distribution (with mean of 0 and the standard deviation of 0.01), optimizing the model with Adam algorithm. We tested the batch size of [128, 256, 512, 1024], and the learning rate of [0.0001, 0.0005, 0.001, 0.005]. When evaluating the latent contexts extraction models, we applied and compared both the auto-encoding and the PCA methods for obtaining unstructured latent contexts. We show the best results obtained by the auto-encoding method.

5 Hierarchical Latent Contexts Analysis: A Case Study

As described in Section 3.2, hierarchical latent contexts are learned by recognizing similar latent contextual patterns using hierarchical clustering methods. In order to demonstrate the necessity of modeling the hierarchical structure of latent contexts for the recommendation process, we visualize the unstructured latent contextual space by using the t-SNE method (t-distributed stochastic neighbor embedding) [39] with LDOS-CoMoDa dataset. By applying t-SNE, we show the hierarchical nature of the latent space, which is reduced to two-dimensions. We illustrate that similar latent context points, that are clustered together, can be represented by a particular contextual situation derived from our hierarchical latent representation model. Moreover, we can also show the importance of aggregating clusters in a hierarchical manner for better representing high-level contextual situations.

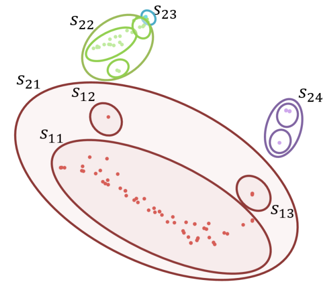


Fig. 4. T-SNE graph of the latent context's vector space with hierarchical contextual situations applied for LDOS-CoMoDa dataset.

The output graph obtained from applying t-SNE contains a set of clustered points, each of them representing a latent contextual vector, as presented in Figure 4. A different color is used for each cluster (cluster 1- red, cluster 2- green, cluster 3- blue, and cluster 4- purple). Separate from generating the t-SNE graph, we constructed a hierarchical latent contextual tree (as explained in Section 3.2 and shown in Figure 5) in order to extract hierarchical latent contexts for each latent contextual vector. Then, we matched each latent contextual point from the t-SNE graph to its extracted hierarchical latent contextual situation $s_{h,i}$. As shown in Figure 4, we marked groups of latent contextual instances that belong to the same contextual situation in a circle and named each group with its contextual situation.

Although the training process of the hierarchical tree was done on all the latent contextual vectors of LDOS-CoMoDa dataset (2,294 ratings, as presented in Section 4.1), we used only two randomly selected users that rated 102 movies in total because we would like to comprehend the latent contextual space and its hierarchical structure using only few (102) examples that can be clearly visualized. As is shown in Figure 5, the partial tree has 102 leaves (shown as colored circles) representing the extracted latent contextual vectors for the two users, each latent contextual vector having 12 latent attributes. Then, we applied t-SNE with the latent contextual vectors to reduce their dimensionality to a two-dimensional space, thus producing a t-SNE graph with two-dimensional latent contextual points. We added the corresponding contextual situations from the learned tree (marked as a circle and named as $s_{h,i}$ in Figure 4) for each point in the t-SNE graph.

It can be seen from Figure 4 that similar latent contextual vectors, that are close to each other, are grouped into a similar contextual situation. This illustrates our assumption, expressed in Section 3.2, that similar contextual patterns can be revealed by clustering unstructured latent contextual vectors. In addition, we can observe from Figure 4 the

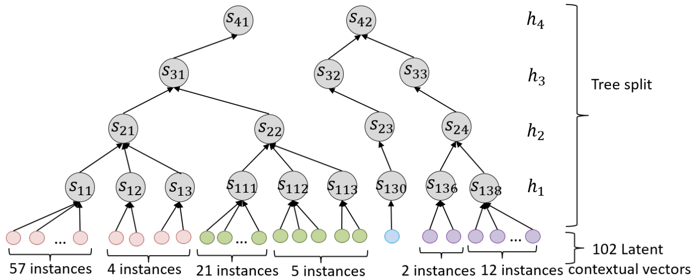


Fig. 5. Example of the hierarchical latent contextual model applied for the LDOS-CoMoDa dataset.

Contextual Situation	Original Contextual Values											
Situation	Interaction	Decision	Physical	Mood	Dominant Emotion	End Emotion	Social	Weather	Location	Season	Day type	Time
S11	First	Given	Healthy	Neutral	Neutral	Neutral	Alone	Sunny	Home	Autumn	Working	Evening
S13	First	Decision	Healthy	Neutral	Happy	Neutral	Partner	Cloudy	Home	Autumn	Weekend	Night
S112	First	Given	Healthy	Positive	Neutral	Happy	Partner	Rainy	Friend	Autumn	Working	Evening
S111	First	Decision	Healthy	Neutral	Happy	Happy	Partner	Rainy	Friend	Autumn	Working	Evening
S130	First	Decision	Healthy	Negative	Happy	Happy	Partner	Cloudy	Friend	Autumn	Weekend	Evening
S138	Nth	Given	Ill	Negative	Neutral	Neutral	Family	Cloudy	Friend	Winter	Holiday	Night
S136	Nth	Given	Ill	Neutral	Neutral	Neutral	Family	Cloudy	Friend	Winter	Holiday	Night

Fig. 6. Original contextual features and their corresponding learned contextual situations.

hierarchical nature of latent contextual vectors, which can be grouped into several contextual situations at different granularity levels. For example, from Figures 4 and 5 we can see the red cluster, which corresponds to our extracted contextual situation s_{21}. s_{21} contains 61 latent contextual instances that are divided into three lower level contextual situations: s_{11}, s_{12}, and s_{13}, each of them contains 57, 2, and 2 instances respectively. We can also notice that a single contextual instance (marked as blue in Figure 4) was grouped in our hierarchical tree to a contextual situation s_{23}, although this point in the t-SNE graph is very close to the green instances that belong to the contextual situation s_{22}. This phenomenon can be explained by the fact that the t-SNE graph contains only a part of the latent contextual instances extracted from the LDOS-CoMoDa dataset. Since we chose to show only the latent contextual instances of two users, the instances of other users may share other contextual patterns with this instance.

In order to further investigate this phenomenon and better understand the meaning of each contextual situation, we show the original contextual feature space for each extracted contextual situation, as shown in Figure 6. Since for each contextual situation there are multiple instances (i.e., s_{12} contains 4 instances), we randomly selected one instance from the dataset that represents each contextual situation. We can notice from Figures 5 and 6 that both of the red (s_{11} and s_{13}) and green (s_{112} and s_{111}) situations, which were clustered to a higher-level contextual situation s_{31}, share "Happy" and "Neutral" moods, while watching a movie, which confirms that a meaningful contextual situation was revealed by our structural model of the hierarchical tree. In addition, we can notice the main differences between their corresponding higher-level contextual situations s_{21} and s_{22}: the red cluster s_{21} that contains s_{11} and s_{13} and the green cluster that contains s_{112} and s_{111} are different in their location and the end emotion values. Hence, different emotions and locations differentiate between the learned contextual situations s_{21} and s_{22}. We can also notice that the blue instance, which belongs to situation s_{130}, has a negative mood, as opposed to the green instances s_{112} and s_{111} that have "Positive" or "Neutral" moods. This further highlight the necessity of the hierarchical structure, which can reveal unique patterns within the latent contextual values. As we can also notice from Figure 5, the blue (s_{130}) and the purple (s_{138} and s_{136}) clusters were grouped into the same higher-level contextual situation s_{42} in the 4th level of the hierarchical tree presented in Figure 5. This is probably because most of their instances share negative moods, as shown in Figure 6.

We can also notice the dimensionality difference between latent contextual vectors and hierarchical latent vectors. While the unstructured latent contextual vectors were extracted by an auto-encoding method and were represented by 12 latent values (out of 32 normalized and binary contextual features), the hierarchical latent vectors were modeled by 4 contextual situations that were automatically obtained from the path of each latent contextual vector in the tree, as explained in Section 3.2. Therefore, the representation of hierarchical latent vectors is more compact than the latent vector representation and can help better find meaningful contextual situations in latent contexts.

6 Results

We present the rating prediction results of our two proposed methods HLCM and Hybrid for the high-, medium- and low-dimensional context-aware datasets, using two splitting strategies: 10-fold cross-validation (CV) and time-based splitting strategy. Table 2 illustrates the rating prediction performance with respect to RMSE and MAE on two *high-dimensional* context-aware datasets, Hearo and CARS, that contain 523 and 247 contextual features respectively. The best results in each column are denoted in bold, while results that are statistically significant ($p < 0.05$) are denoted by an asterisk (*).

Table 2 shows that our new representation of hierarchical latent contexts model (HLCM) consistently achieves the best performance on both metrics and on both datasets with respect to the other baselines. From the table, we have the following observations. First, context-aware models outperform context-unaware models MF and IHSR, which confirms the fact that contextual information is an important factor to recommender systems. Second, we can notice that the models of factorization machines (FM) and Convolutional FM (CFM) achieved better results than most of existing context-aware models (ECM and LCM) for the CARS dataset. This can be explained by the fact that CARS dataset has 38,900 ratings, while Hearo dataset has only 7,416 ratings. Hence, when relevant data exists, FM and CFM can be trained more accurately to find high-order

TABLE 2
Results of Rating Prediction on High-Dimensional Context Datasets

	Hearo Dataset				CARS Dataset			
	10-fold CV		Time-based		10-fold CV		Time-based	
Model	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
MF	0.441	0.385	0.461	0.412	0.489	0.397	0.509	0.415
$IHSR$	0.43	0.379	0.452	0.391	0.467	0.373	0.484	0.396
FM	0.433	0.377	0.454	0.393	0.356	0.28	0.38	0.301
CFM	0.429	0.359	0.451	0.386	0.35	0.274	0.371	0.296
ECM	0.428	0.332	0.449	0.348	0.395	0.301	0.412	0.314
LCM	0.426	0.324	0.442	0.335	0.381	0.319	0.394	0.33
$HLCM_{MF}$	0.421	0.321	0.436	0.331	0.315	0.24	0.337	0.26
$HLCM_{IHSR}$	0.418	0.319	0.434	0.328	0.314	0.238	0.331	0.256
$Hybrid_{MF}$	0.402*	0.306*	0.418*	0.309*	0.312*	0.231*	0.329*	0.242*
$Hybrid_{IHSR}$	**0.392***	**0.301***	**0.410***	**0.3***	**0.309***	**0.229***	**0.325***	**0.239***

TABLE 3
Results of Rating Prediction on a Medium-Dimensional Context Dataset

LDOS-CoMoDa Dataset		
Model	RMSE	MAE
MF	1.45	1.1
$IHSR$	1.43	1.08
FM	1.4	1.09
CFM	1.4	1.08
ECM	1.43	1.1
LCM	1.41	1.09
$HLCM_{MF}$	1.38	1.07
$HLCM_{IHSR}$	1.36	1.05
$Hybrid_{MF}$	1.26*	0.99*
$Hybrid_{IHSR}$	**1.23***	**0.97***

correlations and can significantly improve recommendation accuracy. Third, our hierarchical latent contexts model consistently outperforms all other baselines in both variations of MF and IHSR. In particular, the extension of IHSR achieves significantly better results: in the 10-fold CV setting, $HLCM_{IHSR}$ obtains improvements over the best baseline by 1.9% in RMSE and 1.6% in MAE on Hearo dataset. On CARS dataset, $HLCM_{IHSR}$ improves over the best baseline CFM by 11.5% and 15.1% in terms of in RMSE and MAE respectively in the 10-fold CV setting, and 12.1% and 15.6% in RMSE and MAE respectively in the time-based setting. This observation confirms our assumption that the hierarchical representation better captures latent contextual information than the unstructured vector-based approach, since it identifies complex contextual situations that improve recommendation performance.

Finally, we can observe that the hybrid approach described in section 3.4, which combines hierarchical and unstructured latent contexts vectors, leads to substantial performance improvements in context-aware recommendations and significantly outperforms all other baselines. The hybrid approach obtains improvements over the best baseline $HLCM_{IHSR}$ by 6.6% in RMSE and in 5.98% MAE on Hearo dataset in the 10-fold CV setting, and 5.85% and 9.33% in terms of RMSE and MAE respectively in the time-based setting. On CARS dataset the hybrid approach improves over the best baseline by 1.61% in RMSE and 3.93% in MAE in the 10-fold CV setting, and similar results were obtained in the time-based setting. This observation indicates the importance of incorporating two types of latent contextual representations into the recommendation model; the first is the actual value of the context provided by the unstructured latent contextual vector, and the second is the contextual situations at different granularity levels that are provided by the hierarchical latent contexts.

In Table 3, we present the rating prediction results on the *medium-dimensional* context-aware dataset LDOS-CoMoDa, which contains 12 contextual dimensions. We can observe that our proposed approaches of HLCM and Hybrid that extend the IHSR model achieve significantly better performance than all the other baseline models. Therefore, it further confirms the usefulness of hierarchical latent contexts for better recommendations. In addition, the rating performance of CFM was superior to the existing context-aware models, as it outperforms ECM and LCM models and better captures higher-order feature interactions. Another major contribution of hierarchical latent contexts can be seen in this medium-dimensional setting, where the hybrid approach performed much better than the HLCM. Specifically, in the 10-fold CV setting the hybrid approach that extends IHSR ($Hybrid_{IHSR}$) improves the best baseline CFM by 13.8% and 11.3% in terms of RMSE and MAE respectively. This further verifies the substantial influence of combining hierarchical and unstructured latent contexts together. Therefore, we can conclude that in settings when contextual dimension is medium or high, unstructured latent contextual vectors and hierarchical contextual situations are *complementary to each other* and result in the best rating performance.

In order to further highlight the impact of hierarchical latent contexts on the results accuracy, we examined how limited number of contextual dimensions affected the recommendation accuracy, as shown in Table 4. We evaluated our hierarchical latent context model on three *low-dimensional* datasets: Spotify, Frappe and Yelp, which contain 5, 3 and 2 contextual dimensions respectively. The results show that the hybrid model extending the IHSR model consistently achieves the best performance on both metrics and on all datasets with related to the other baselines. From Table 4 we have three main observations. First, we can observe that our proposed approach of $HLCM_{IHSR}$ utilizing only hierarchical latent contexts obtained slightly better results than LCM and ECM on the Frappe dataset. This phenomenon can be explained by the fact that reducing the dimension of the context causes simpler and limited number of learned contextual situations. Hence, when we extract hierarchical latent contexts from low-dimensional contextual space, the hierarchical information can become sparse and uninformative since it represents the recurrent latent contextual patterns. Second, on the Yelp and Spotify datasets, adding hierarchical latent contexts in the $HLCM_{IHSR}$ model improves the other baselines in terms of MAE and RMSE for both splitting methods. This observation confirms our assumption that in big-scale environments, complex contextual situations can be learned efficiently, even when context space is low. In addition, we can observe the impact of the application of deep learning on the recommendation results, as in most of the cases CFM achieves the best performance with related to the other context-aware baselines. Third, we can observe that the hybrid approach, utilizing both hierarchical and unstructured latent contexts, improves HLCM approach in terms of all rating measures. Specifically, in the time-based

TABLE 4
Results of Rating Prediction on Low-Dimensional Context Datasets

	Spotify Dataset				**Frappe Dataset**		**Yelp Dataset**			
	10-fold CV		**Time-based**		**10-fold CV**		**10-fold CV**		**Time-based**	
Model	RMSE	MAE	RMSE	MAE	RMSE	MAE				
MF	0.287	0.258	0.293	0.261	0.786	0.624	1.128	0.889	1.16	0.89
$IHSR$	0.28	0.251	0.285	0.254	0.764	0.61	1.11	0.875	1.148	0.885
FM	0.262	0.237	0.268	0.242	0.71	0.55	1.109	0.872	1.141	0.881
CFM	0.259	0.22	0.265	0.224	0.7	0.53	1.1	0.87	1.135	0.876
ECM	0.265	0.239	0.27	0.243	0.69	0.523	1.11	0.876	1.146	0.883
LCM	0.259	0.221	0.264	0.223	0.689	0.522	1.103	0.869	1.132	0.871
$HLCM_{MF}$	0.254	0.218	0.262	0.22	0.691	0.526	1.098	0.859	1.131	0.868
$HLCM_{IHSR}$	0.249	0.212	0.259	0.218	0.688	0.524	1.092	0.853	1.128	0.865
$Hybrid_{MF}$	0.248*	0.21*	0.259*	0.219*	0.688*	0.52*	1.077*	0.846*	1.09*	0.851*
$Hybrid_{IHSR}$	0.238*	0.197*	0.246*	0.209*	**0.686***	**0.51***	**1.074***	**0.841***	**1.08***	**0.85***

splitting setting the hybrid approach $Hybrid_{IHSR}$ obtains improvements over the best baseline LCM by 4.8% in RMSE and 2.47% in MAE on the Yelp dataset, 8.8% in RMSE and 12.18% in MAE on the Spotify dataset, and on the Frappe dataset the improvement of the hybrid approach is negligible. Similar results were obtained in the 10-fold CV setting. This observation indicates that when the contextual space is small, utilizing both latent and hierarchical latent contexts in the hybrid approach can still produce comparable results and significantly improve recommendation accuracy.

7 Conclusion

In this paper we proposed a novel representation of latent contextual information defined in a hierarchical manner for recommender systems. This hierarchical latent representation captures latent contextual information better than the unstructured vector-based approach since it identifies the patterns of latent contextual information, referred as contextual situations, that are useful for the conceptual understanding of latent contexts. This representation is more compact than the latent vector representation and can help better find meaningful contextual situations and can overcome the sparsity problem of context-aware recommender systems.

We presented two novel latent context-aware recommendation models that extend traditional CF-based models by utilizing latent context information in a hierarchical and unstructured manner. The first is a contextual recommendation model that incorporates hierarchical latent contexts (HLCM) and the second is a hybrid contextual recommendation model that utilizes hierarchical and unstructured latent contexts in a complementary manner. We conducted several experiments to compare our approaches to state-of-the-art recommendation models, with respect to RMSE and MAE measures.

Experimental results on six context-aware datasets showed that our models significantly outperform baseline context-aware methods used for rating prediction task. Specifically, we showed that the suggested recommendation model utilizing hierarchical latent contexts (HLCM) was superior to the latent context baseline model (CFM) in terms of the RMSE and MAE metrics, especially for the medium to high-dimensional cases. We also observed that utilizing both hierarchical latent contexts and unstructured latent contexts in a hybrid approach significantly achieves the best performance on all datasets. We demonstrated that for the high-dimensional contextual case, a relatively small size of contextual situations is what you actually need to make the recommendation process effective, while for settings with medium or low-dimensional context, it is preferable to utilize both latent context and hierarchical latent context for improving recommendation accuracy.

We believe that the explainabilty of the hierarchical model is one of the limitations of our work. Since the construction of hierarchical latent contexts is done in a latent space with poorly interpretable numerical data, labeling and annotating latent contextual situations is a complex task. Unlike topic modeling and similar problems in explicit feature spaces, such interpretation still remains challenging and requires conducting extensive additional research.

The hierarchical latent contextual representation can be important beyond its applicability in recommender systems and can be used for other domains. For example, it can be used by marketers to understand complex contexts of customer actions and by data scientists in general for developing better predictive models, such as predicting if a customer would respond to a mobile phone's ad in the contextual situation of him entering a shopping mall on Saturday evening with his girlfriend. In addition, since it still remains unclear if improvements on rating prediction measures like RMSE translate into more effective recommendations, it is also important to measure the impact of supporting hierarchical latent contexts in recommender systems using business performance metrics, such as CTR, adoption and conversion rates, and increased sales and revenues, as opposed to using classical ML metrics, such as RMSE, F-measure, etc. We plan to work on these issues as part of our future research.

References

[1] G. Adomavicius and A. Tuzhillin, "Context-aware recommender systems," in *Recommender systems handbook*. Springer, 2011, pp. 217–253.

[2] L. Mei, P. Ren, Z. Chen, L. Nie, J. Ma, and J.-Y. Nie, "An attentive interaction network for context-aware recommendations," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 157–166.

[3] Y. Zheng, B. Mobasher, and R. Burke, "Cslim: Contextual slim recommendation algorithms," in *Proceedings of the 8th ACM Conference on Recommender Systems*. ACM, 2014, pp. 301–304.

[4] X. Xin, B. Chen, X. He, D. Wang, Y. Ding, and J. Jose, "Cfm: convolutional factorization machines for context-aware recommendation," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 3926–3932.

[5] M. Okawa, T. Iwata, T. Kurashima, Y. Tanaka, H. Toda, and N. Ueda, "Deep mixture point processes: Spatio-temporal event prediction with rich contextual information," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 373–383.

[6] X. Ding, J. Tang, T. Liu, C. Xu, Y. Zhang, F. Shi, Q. Jiang, and D. Shen, "Infer implicit contexts in real-time online-to-offline recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2336–2346.

[7] M. Unger, A. Bar, B. Shapira, and L. Rokach, "Towards latent context-aware recommendation systems," *Knowledge-Based Systems*, vol. 104, pp. 165–178, 2016.

[8] W. Wu, J. Zhao, C. Zhang, F. Meng, Z. Zhang, Y. Zhang, and Q. Sun, "Improving performance of tensor-based context-aware recommenders using bias tensor factorization with context feature auto-encoding," *Knowledge-Based Systems*, vol. 128, pp. 71–77, 2017.

[9] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[10] M. Hayat, M. Bennamoun, and S. An, "Learning non-linear reconstruction models for image set classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1907–1914.

[11] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.

[12] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Systems with Applications*, vol. 105, pp. 233–261, 2018.

[13] K. Haruna, M. Akmar Ismail, S. Suhendroyono, D. Damiasih, A. Pierewan, H. Chiroma, and T. Herawan, "Context-aware recommender system: A review of recent developmental process and future research direction," *Applied Sciences*, vol. 7, no. 12, p. 1211, 2017.

[14] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[15] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[16] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 635–644.

[17] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 2011, pp. 301–304.

[18] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, p. 5, 2019.

[19] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, "A review on deep learning for recommender systems: challenges and remedies," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 1–37, 2019.

[20] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 355–364.

[21] Y. Jhamb, T. Ebesu, and Y. Fang, "Attentive contextual denoising autoencoder for recommendation," in *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*. ACM, 2018, pp. 27–34.

[22] E. Smirnova and F. Vasile, "Contextual sequence modeling for recommendation with recurrent neural networks," *arXiv preprint arXiv:1706.07684*, 2017.

[23] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 2016, pp. 233–240.

[24] S. Wang, J. Tang, Y. Wang, and H. Liu, "Exploring hierarchical structures for recommender systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1022–1035, 2018.

[25] L. Pradhan, C. Zhang, and S. Bethard, "Infusing latent user-concerns from user reviews into collaborative filtering," in *2017 IEEE International Conference on Information Reuse and Integration (IRI)*. IEEE, 2017, pp. 471–477.

[26] J. Han, L. Zheng, H. Huang, Y. Xu, S. Y. Philip, and W. Zuo, "Deep latent factor model with hierarchical similarity measure for recommender systems," *Information Sciences*, vol. 503, pp. 521–532, 2019.

[27] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 1, p. 1, 2009.

[28] N. D. Lane and P. Georgiev, "Can deep learning revolutionize mobile sensing?" in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*. ACM, 2015, pp. 117–122.

[29] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1971–1980.

[30] C. Liu, L. Zhang, Z. Liu, K. Liu, X. Li, and Y. Liu, "Lasagna: towards deep hierarchical understanding and searching over mobile sensing data," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 334–347.

[31] M. A. Domingues, M. G. Manzato, R. M. Marcacini, C. V. Sundermann, and S. O. Rezende, "Using contextual information from topic hierarchies to improve context-aware recommender systems," in *2014 22nd International Conference on Pattern Recognition*. IEEE, 2014, pp. 3606–3611.

[32] M. Unger, B. Shapira, L. Rokach, and A. Livne, "Inferring contextual preferences using deep encoder-decoder learners," *New Review of Hypermedia and Multimedia*, vol. 24, no. 3, pp. 262–290, 2018.

[33] I. Davidson and S. Ravi, "Agglomerative hierarchical clustering with constraints: Theoretical and empirical results," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2005, pp. 59–70.

[34] F. Murtagh and P. Legendre, "Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion?" *Journal of classification*, vol. 31, no. 3, pp. 274–295, 2014.

[35] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.

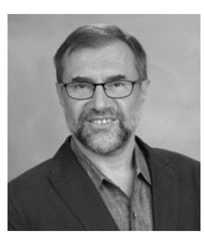
[36] M. Unger, B. Shapira, L. Rokach, and A. Bar, "Inferring contextual preferences using deep auto-encoding," in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM, 2017, pp. 221–229.

[37] A. Odic, M. Tkalcic, J. F. Tasic, and A. Košir, "Relevant context in a movie recommender system: Users opinion vs. statistical detection," *ACM RecSys*, vol. 12, 2012.

[38] L. Baltrunas, K. Church, A. Karatzoglou, and N. Oliver, "Frappe: Understanding the usage and perception of mobile app recommendations in-the-wild," *arXiv preprint arXiv:1505.03014*, 2015.

[39] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

Moshe Unger received the PhD degree in 2018 from Ben-Gurion University. He is currently a postdoctoral research scientist at the Leonard N. Stern School of Business, NYU, where he conducts applied machine learning research in the field of recommender systems. His research interests include data mining, personalization and deep learning.



Alexander Tuzhilin is Leonard N. Stern Professor of Business at the Stern School of Business, NYU. His research interests include personalization, recommender systems, machine learning and AI. He has produced over 130 research publications on these and related topics. Professor Tuzhilin has served on the organizing committees of numerous conferences, including as the Program and as the General Chair of the IEEE International Conference on Data Mining (ICDM), and as the Program and the Conference Chair of the ACM Conference on Recommender Systems (RecSys). He served on the editorial boards of several journals, including as the Editor-in-Chief of the ACM Transactions on Management Information Systems. He received his PhD in Computer Science from the Courant Institute of Mathematical Sciences at NYU.