Computer Science Review 31 (2019) 84–97

Contents lists available at ScienceDirect

Computer Science Review

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

Progress in context-aware recommender systems — An overview☆

Shaina Raza *, Chen Ding

Ryerson University, Toronto, Canada

ARTICLE INFO

Article history:
Received 27 July 2018
Received in revised form 10 January 2019
Accepted 23 January 2019
Available online 1 February 2019

Keywords:
Context
Recommender systems
Context-aware
Dimensionality reduction
Contextual modeling
User modeling

ABSTRACT

Recommender Systems are the set of tools and techniques to provide useful recommendations and suggestions to the users to help them in the decision-making process for choosing the right products or services. The recommender systems tailored to leverage contextual information (such as location, time, companion or such) in the recommendation process are called context-aware recommender systems. This paper presents a review on the continual development of context-aware recommender systems by analyzing different kinds of contexts without limiting to any specific application domain. First, an in-depth analysis is conducted on different recommendation algorithms used in context-aware recommender systems. Then this information is used to find out that how these techniques deals with the curse of dimensionality, which is an inherent issue in such systems. Since contexts are primarily based on users' activity patterns that leads to the development of personalized recommendation services for the users. Thus, this paper also presents a review on how this contextual information is represented (either explicitly or implicitly) in the recommendation process. We also presented a list of datasets and evaluation metrics used in the setting of CARS. We tried to highlight that how algorithmic approaches used in CARS differ from those of conventional RS. In that, we presented what modification or additions are being applied on the top of conventional recommendation approaches to produce context-aware recommendations. Finally, the outstanding challenges and research opportunities are presented in front of the research community for analysis

Contents

1. Introduction 84
2. Methodology for paper selection 86
3. General perspective of algorithms in context-aware recommender systems 86
4. Dimensionality reduction techniques used in context-aware recommender systems 89
5. User modeling techniques used in context-aware recommender systems 90
6. Evaluation metrics and datasets in context-aware recommender systems 92
7. Discussion and future research directions 94
8. Conclusion 94
Acknowledgments 95
References 95

1. Introduction

The exponential growth of data in the electronic world ranges far beyond the gigabytes, exabytes or petabytes to any volume that may be unimaginable. The concept of Big Data is evolving continuously and is the driving force behind the revolution of digital transformation including Data Science, Internet of Things and Data Mining. This data is so big that it cannot be handled by conventional database systems; in fact we need specialized tools for analyzing and gaining insights from this data [1].

Recommender Systems (RS) are the tools designed for interacting with voluminous and complex information spaces, and prioritizing those items for the users that may be of interest to them [2]. Abstractly speaking, RS consists of techniques used to provide suggestions for any type of data to the users to aid them in the decision-making processes and add a lot to the overall user experience. The ability of a RS to reduce the information bulk from

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.cosrev.2019.01.001.

* Corresponding author.
E-mail addresses: shaina.raza@ryerson.ca (S. Raza), cding@ryerson.ca (C. Ding).

https://doi.org/10.1016/j.cosrev.2019.01.001
1574-0137/

a huge amount of data to provide personalized services to its users has led to the development of many real-world applications [3]. The evidence of its applications can be seen by its acceptance across various domains such as e-commerce, e-learning, e-government, e-library, e-business services and its coverage in the broad range of applications like music, movies, websites, tourism, books, documents, research articles, food and other products in general [3]. The early research in RS evolved as result of research in information retrieval and message filtering [4]; and RS has emerged as an independent research field in the mid-90s with researchers' focus shifted towards various recommendation problems that mainly rely on the rating structure [2]. Commonly used recommendation approaches include collaborative filtering, content-based filtering, demographic based, utility based, knowledge based and hybrid recommendation systems [2].

Traditionally, recommendation approaches are based on two dimensions (users $\times$ items) where a set of N items are expected to be of interest to M users [2]. Over the time, it had been observed that the recommendation quality of these traditional RS is quite low due to the homogeneity of the information sources and insufficient user/items data. To handle this, the research community in the early 2000s began investigating the notion of *context* in the recommendation approaches. The information to characterize the ongoing situation of an entity is called *context* [5]. This has given birth to a new kind of RS known as Context-Aware Recommender System (CARS) [6]. In CARS, the classical two-dimensional process is extended to leverage the contextual information to provide better personalized recommendations to its users.

There has been some prior research in context-aware systems and the researchers are becoming more interested to work in this field since the term context-aware recommender system (CARS) had been coined for the first time in the literature [6]. However, we have found very few survey papers such as [7–9] that have conducted thorough literature review to trace the development in CARS. Adomavicius and Tuzhilin had refined their own work in CARS from the year 2008 to 2015 [7]. In their work, they explained the concept of context [5] in detail, introduced the paradigms for incorporating context in a RS (contextual pre-filtering, modeling and post-filtering), and explained the contextual extraction methods that can be used in CARS.

In one of the initial surveys, Hong et al., [10] reviewed the publications from the year 2000 to 2007 on context-aware systems to review the development in the field. Similarly, Truong and Dustdar [11] presented a survey on context-aware systems covering different web services. Though not directly related to CARS, there are several lessons that can be learnt from these surveys, including different classification layers (concept, network, middleware, application and user layers) and emerging issues in context-aware systems. Verbert [12] presented a survey on Technology Enhanced Learning (TEL) RS in which they identified appropriate contextual dimensions for TEL applications and then analyzed the existing systems for these contexts. A survey that reviewed different tools and techniques to develop personalized music services for the users in context-aware music retrieval and recommendation services had been conducted in 2012 [13]. A focused survey had been presented on context-aware mobile recommendations [14] in which the authors first discussed the notion of mobile context and then discussed context gathering and reasoning techniques in this domain.

Different from the preceding surveys, a survey on CARS [15] compared contextual pre-filtering, post-filtering and contextual modeling methods in terms of accuracy and diversity of a RS to find out which one performs better than the other under certain settings. A review had been conducted on the related CARS papers from the year 2001 to 2013 for the subject scholarly RS [16] to identify the contexts and techniques used for making effective recommendation in digital libraries. There is a survey on social TV recommender systems [17] in which authors analyzed context-awareness in making predictions according to viewers' behavior.

A literature review on CARS based on computational analysis techniques [18] was conducted in which the authors first presented a taxonomy of computational analysis techniques and then classified the publications based on these techniques. There is another survey that reviewed the state-of-the-art CARS in mobile environment [19] to classify them into different sub-systems based on location, time, activity, social, emotion or otherwise multiple dimensions. Ubiquitous computing is one of the emerging field around RS that is built on the top of wireless communication and sensors. A survey on the context-aware ubiquitous learning systems [20] was conducted to review various learning perspectives in terms of context-awareness and to propose further refinement in the field.

An integrated approach to CARS development has been recently presented in a literature review [8] on the subject from the year 2010 to 2017. They compared the latest work based on the application domains being used, filtration techniques (pre-filtering, contextual modeling and post-filtering), extraction methods (contextual information is extracted explicitly, implicitly or by statistical means to build user models) and evaluation approaches (online, offline, user studies with different evaluating metrics) being used. Lately, a survey has been presented on the publications in CARS from 2004 to 2016 [9] in which authors reviewed the distribution of application domains in CARS, different types of contexts used in traditional recommendation approaches; and the contextual recommendation approaches (pre-filtering, contextual modeling and post-filtering) to incorporate context into RS. They also researched different validation mechanism used in general RS and analyzed their distribution across the CARS applications.

As it can be observed that each of the above surveys has covered one or a few of the specific aspects of the CARS development process. Almost all the surveys have discussed the recommendation algorithms except [15,17]. Only four surveys [7,8,10,15] discuss CARS in general, not limited to a specific type of application domain and context being used. Two surveys [8,9] discuss the datasets and evaluation techniques, but the reviewed papers in these surveys belong to the general RS and the attempt had been made to find out the aspects of context-awareness in the datasets and evaluation metrics. In that, they are not related to pure CARS datasets and evaluation techniques. Panniello et al., [15] has discussed two evaluation metrics: accuracy and diversity for comparing CARS, but those metrics are not specific to CARS. Unlike these surveys, our paper has presented a continual approach to the comprehensive CARS progress. Our literature review has the following criteria filled up that differentiate it from other survey papers:

- Our survey is not domain dependent and not focused on one type of context, unlike the surveys that focus on pedagogy [12,16], on music [13,19], on mobile [14], on TV recommendations [17] and on computational intelligence techniques [18].
- We have highlighted some of the most favorable and important algorithms (in Section 2) that are crucial for building CARS application and they were not discussed in other surveys.
- None of the surveys has discussed dimensionality reduction in their surveys. Adomavicius and Tuzhilin [7] had touched this topic, but we have significantly explored this topic by reviewing different algorithms that how they tackle this issue.
- We have highlighted the subject of user modeling, datasets and evaluation protocols specifically used in CARS. Besides the discussion initiated in [8,9], we further reviewed the state-of-the-art in CARS to find out more about aspects of context-awareness from different authors.

By doing so, we aim to fill the current gap in the state-of-the-art research in CARS and to provide background knowledge to both the novice and expert researchers in this field. We have classified the literature based on the algorithmic approaches being used in CARS and reviewed them in depth on how they tackle the inherent issues in CARS like curse of dimensionality [21]; how user preferences are incorporated in the recommendation process; why there is lack of context-aware datasets and what validation methods are used in the current state-of-the-art.

The remaining sections in this paper are organized as follows. Section 2 explains the methodological aspects to extract relevant literature. Section 3 presents the classification of algorithmic approaches. Section 4 is about the dimensionality reduction techniques. Section 5 describes user modeling techniques. Section 6 is about the datasets and evaluation techniques used in CARS. In Section 7, we provide a discussion and conclude the paper in Section 8.

2. Methodology for paper selection

We identified and selected the following collections of bibliographies for literature in CARS based on the quality of the publications and their relevancy on the subject : ACM Digital Library, SpringerLink, IEEE Xplore and Elsevier. We also identified the following scholarly search engines to look for the pertinent literature: GoogleScholar, DBLP, CiteSeerX, Microsoft Academic Search, Web of Science, ScienceDirect and ResearchGate.

We also browsed the conference proceedings and journal transactions to look for the titles and the abstracts for finding more papers which might have been initially skipped in the earlier search. We had specified January 2008 as the starting date based on the birth of the subject [6] and December 2017 as the closing date for our literature review. Besides this specified time frame, we have also cited a few classical and some latest publications for deeper study.

We used the Boolean search query ((“Context-aware” OR “contextual” OR “Contextualization”) AND (“Recommender System” OR “Recommendation System” OR “Recommendations”) to search the bibliographies with the following inclusion criteria: (i) papers written in English and (ii) relevancy and usefulness to the topic. We excluded the papers from workshops, symposium proceedings and those with fewer than 10 citations. After these filters, we got around 110 papers from the data extraction strategy. We thoroughly analyzed these papers to look for the development of CARS. This filtration process has returned us about 58 papers, out of which 45 are those manuscripts which are proposing CARS and 13 are the survey papers.

3. General perspective of algorithms in context-aware recommender systems

Earlier work in CARS has been performed at the algorithmic level where the researchers have tried to make use of contextual information to improve existing algorithms [7]. Research into this issue has produced many variants of the traditional RS algorithms that are being used in CARS. During our preliminary studies on CARS, we found out that there is still a lot of scope for defining the notion of *context* [5,22] and there needs to be more rigorous research to model such information in the RS . Doing so, we can present an opportunity to the academics to demonstrate the best working algorithms in building CARS applications.

Contexts can be incorporated at various stages of the recommendation process including contextual pre-filtering, contextual post-filtering and contextual modeling [7]. In contextual pre-filtering and contextual post-filtering, the ratings are predicted using any two-dimensional rating function. Context is used in advance in pre-filtering to construct the data records or ratings, whereas in post-filtering context is applied on the resulting set of recommendations. Contextual modeling applies contextual information directly as part of the rating estimation. In this section we have discussed our chosen publications in terms of how they incorporate the contextual information and the recommendation approach they use in CARS.

Matrix Factorization (MF) [23] is considered to be a de-facto recommendation algorithm which gained first recognition and broader exposure in RS with Netflix Prize [24] for movie recommendation. As the name suggests, MF is the way of decomposing a matrix into product of two matrices, such that original matrix is retained when multiplied. In RS, MF can be used to discover latent features that exhibit interactions between two different kinds of entities (users and items). Given a set M of users and set N items, R is a rating matrix of size $|M| \times |N|$, we can discover K latent features by finding two matrices P(|M| x K) and Q (|N| x K) such that their approximate is $R \approx P \times Q^T$ [25].

The observed variations in rating values is much due to the effects associated with either the users or the items known as biases [26]. Thus, if a standard MF has a formula $r_{ij} = p_i q_j^T = \sum_{i,j}(r_{ij} - p_i q_j^T)$ (r_{ij} is dot product of user vector p_i and item vector q_j) [26] , then a biased MF has a rating $r_{ij} = p_i q_j^T + \mu + b_i + b_j$ [26] (where μ is the global mean rating; b_i and b_j are user and item biases respectively). In context-aware matrix factorization (CAMF), we replace the item bias b_j by the interaction of item i and k contextual conditions $c_i \ldots . c_k$ [26]. The basic prediction rule in CAMF [27] with latent factor vectors for the given user p_i and item q_j ; average items ratings μ and the rating bias B_{ijc_j} for a given item under a contextual condition is defined as:

$$\hat{r}_{i,j,c1..,c_k} = p_i . q_j{}^T + \mu + b_i + \sum_{j=1}^{k} B_{ijc_j} \tag{1}$$

In one of the initial attempts to contextualize MF methods, there are three CAMF models derived based on the interaction of items and contextual factors i.e. CAMF-C (contextual condition independent of item), CAMF-CI (item–contextual condition pair) and CAMF-CC (one model parameter for each contextual condition–item category) for recommending place of interest and music to the users [27]. Students' preferences over questions were incorporated as the item–contextual conditions in the form of item bias in basic CAMF model (Eq. (1)) [28].

The basic CAMF methods [27] represents the non-probabilistic linear model that factorizes the rating matrix into a product of two smaller latent matrices (users and items) with contextual features. The probabilistic matrix factorization (PMF), on the other hand, describes each user and item by a small set of attributes into n-dimensional latent vectors that are not known ahead of time but are discovered by the learning algorithm automatically [29]. PMF had been used to learn various social contexts as latent vectors for the review helpfulness rating prediction problem [30]; online social recommendation problem [31]; and point-of-interests recommendations [32,33]. SLIM (Sparse Linear Method) is traditional MF approach for top-N recommendations, that aggregates sparse ratings on other items rated by a user, to deal with high sparsity and to reduce model learning time [34]. SLIM had been extended to CSLIM (Contextual Sparse Linear Method) [35] by utilizing basic CAMF models [27] to include itemKNN (Item-based k-nearest-neighbor) and userKNN (user-based k-nearest-neighbor) as item and user' biases respectively. Heuristics had also been used to switch between basic CAMF models and their variants [27] to tackle the encountered cold-start situation in CARS (Braunhofer, 2014). Standard MF and CAMF models typically predict for a single user–item matrix under a single context which had been extended to multiple contexts (context-specific and shared contexts) in CARS [36]. Event recommendations had been addressed by replacing the basic rating of a user to an event in the standard CAMF prediction rule (Eq. (1)) with the rating function of a linear contextual feature model that comprises of a variety of contextual features between the user and the item for a social network [37].

As observed, CAMF finds the latent contexts by considering the interactions between the user and item entities only. Recommendation algorithms designed primarily to operate on matrices cannot handle the ternary relational nature of data like tagging, time or location based scenarios [38]. These systems use a third entity to be modeled in the recommendation process. Generalizing this approach leads to formalize the concept of CARS as $User \times Item \times Context_1 \times Context_2 \times \ldots \times Context_k$ [7]. Tensors can incorporate K different contexts as multifaceted user–item interactions in the recommendation process. **Tensor Factorization (TF)** is an extension of MF techniques to reduce each tensor (n-way array) into lower dimensional feature vector [38]. Well-known TF algorithms are Candecomp/Parafac (CP) and Tucker decomposition. Tucker decomposition decomposes a tensor into M mode matrices and a core tensor through Higher Order Singular Value Decomposition (HOSVD). In CP, a tensor is decomposed into rank-one (outer product of two vectors) components and there is no core tensor as in Tucker [39]. In TF, we generalize the rating observations of MF: $Y \in y^{n\times m}$ into $Y \in y^{n\times m\times c}$ (n: users, m: items, c: contextual variables); such that we get an approximate of Y as $F := UM^T$ where F minimizes a loss function $L(F,Y)$ between observed and predicted values [39] with a regularization term Ω and the context variables [40] as shown in Eq. (2):

$$L(F, Y) + \Omega(F) \tag{2}$$

In one of the initial works to generalize the MF, HOSVD had been used to decompose a N-dimensional tensor into factor matrices $U \in \mathfrak{R}^{n\times d_U}$ (users), $M \in \mathfrak{R}^{m\times d_M}$ (items) and $C \in \mathfrak{R}^{c\times d_C}$ (contexts) and a central tensor $S \in \mathfrak{R}^{d_U\times d_M\times d_C}$ [40] to have a decision function as shown in Eq. (3):

$$F_{ijk} = S \times_U U_{i*} \times_M M_{j*} \times_C C_{k*} \tag{3}$$

Parafac decomposition in its simplest form with vectors: $v_{i1}, v_{i2}, \ldots, v_{ik}$ and "$\otimes$" as the outer product for tensors [39] is referred to as:

$$X \approx \sum_{i=1}^{k} v_{i1} \otimes v_{i2} \otimes \ldots \otimes v_{ik} \tag{4}$$

The baseline equation (Eq. (4)) of the PARAFAC had been reconstructed to reduce the large number of modeling pairwise interactions between the entity E and the corresponding contexts c_1, $c_2 \ldots c_k$ by exploiting pairwise interaction TF (PITF) model [41] with higher order tensor to form the tensor as $X \approx \sum_{i=1}^{k} v_{i1}^{E,C1} \otimes v_{i2}^{E,C2} \otimes \ldots \otimes v_{ik}^{E,Ck}$ [42]. Mean Average Precision (MAP) had been averaged across all users, items and contexts included to modify the CP model (Eq. (4)) to make top-N context-aware recommendations for different types of feedback [43,44].

The baseline HOSVD decomposition (Eq. (3)) had been modified to decompose the original tensor T into corresponding low matrices M where the prediction had been redefined as the Hadamard product of the transposed columns from M as $T = [M_1^T \otimes M_2^T \ldots \otimes M_n^T]$ [45]. This algorithm with its higher order case had the ability to scale linearly with dense tensor and cubically with the number of features. To reduce the number of feature space and computational complexity, the N-way product of this algorithm had been replaced by the sum of dot products between three dimensions (user, item, context) and their respective feature vectors [45] to work like PITF reduction [41].

Contextual overload arises due to the inability of CARS to capture the semantics among the contexts as can be seen in preceding work [40,43,44]. Instead of including contexts as separate dimensions, users and items dimensions are represented under certain contextual factors to address contextual overload issue [46]. The biases in basic CAMF models (Eq. (1)) [27] were substituted by a n-order tensor that combines the user–item entities under each context combinations [46]. A ternary tensor had been built $X \in R^{N\times M\times K}$ (N: users, M topics, K: topics) to make top-N recommendations with partially observed preferences [47] in an expert finding RS.

As observed, the above two approaches i.e. MF and TF are more often used in contextual modeling techniques. **Latent Dirichlet Allocation (LDA),** a widely used algorithm in general RS is equally applicable to contextual pre-filtering and post-filtering approaches. LDA infers two distribution matrices from the data; one matrix being the distribution over topics for a given document and the other one as topic-specific distribution over words in the vocabulary [48]. Experimental results by Stevens [49] showed that LDA and MF are similar in an average case to extract latent topics but LDA has better topic coherence (ability to evaluate topics from large textual collection) compared to a MF model.

LDA is a generative probabilistic model that uses Bayesian statistics to discover latent topics in the corpus through observable variables of the words [48]. In the original LDA model [48], given a set of documents there are some latent topics which are not yet observed. Each document has a distribution over these topics and each topic has a distribution over the words in the vocabulary. These topic proportions are identified to individual words in the documents that leads to topic-keyword distributions. These topic proportions are void of users' preferences and contextual features in the naïve approach. In the context-aware LDA models, the topic distributions (proportions) of the standard LDA model were enriched in several ways such as: by including citation based contexts as the bag-of-words in a citation based RS [50]; by including social contexts as bag of words from the users' mobile logs in a mobile based RS [51]; by inferring users' past preferences for the items in a query driven RS [52]; by including users' personal preferences and the contexts (environmental) in a TV content RS [53]; by inferring multiple contexts (tweets, emails, posts or users' personal attributes) from various items (news, social networking) and digital traces (Facebook, Twitter, emails) corpora in a social based RS [54]; and by extracting contexts as association or co-occurrence relations within the item features from a knowledge base in a semantic CARS [55]. A document is often treated as a user profile with a mixture of latent topics (without contextual features) distributed over the vocabulary in the standard LDA model [48]. User profiles were represented as the conditional item probabilities for a given user under a specific context(s) [52]; as the weighted sum of topic distributions from different contexts [54]; and as multinomial document distributed over a set of latent topics with item features [55] in context-aware LDA models.

As observed the techniques mentioned above do not consider the changing user's opinions and item popularities varying with time. **Markov process** is a model that is capable of being in more than one state and has transitions among those states, such that the states and transitions between them are dependent only on the current state of the system [56]. A Markov model is a stochastic model that models sequential data and randomly changing states where it is assumed that the future states will be derived from the current state; not from the previous states to it. A Markov chain model is a Markov process where we can observe the states directly. In a RS, we can use Markov process to predict what item(s) the users will pick next and is effective for making next prediction [57]. At each step in the recommendation, we are concerned with the sequential criteria and the optimization process to make best choice out of many available options. These Markov based RS models are sometimes called sequence-aware RS or time-aware RS where the main input to such problems is an ordered or timestamped list of users' past interactions [58]. The sequence-aware or time-aware RS can also be referred to as a special case of CARS because it incorporates context in the form of users' short-term actions estimated from their long-term past actions [58]. Hidden Markov Model (HMM) can be used to model user's preferences as a sequence of unobservable contexts and build a latent predictive model to estimate unknown ratings. LDA had been used to get latent topics from frequent song tags which are then processed by kth-order Markov model to capture user's contextual states changing over time [59]. Factorized Markov Decision Process, a sequential session based approach had been used to detect users' goals (or topics) influenced by their actions [60]. Bayesian Hidden Markov Model had been used to address the context recognition of mobile users through probabilistic distributions and transitions of contextual data with prior knowledge of features [61]. Dirichlet Process Mixture and Mixture Unigram model were integrated in the recommendation process to model time sequence and dependency among the latent context topics and the features [61]. Sequential user behavior captured through Markov Chain model had been integrated with the users' preferences based on historical data from a general RS to produce a hybrid model with both the users' and sequential tastes [62]. Hierarchical Hidden Markov modeling had been used to capture the dynamics of changing users' preferences based on current context and collective sequence of historical feedback [63,64].

Though not so popular as the above-mentioned techniques in CARS, **Bandit algorithms** can help to optimize the results when we do not have sufficient data to create a rigorous statistical model. Originally developed by Robbins (1952), the bandit problem is a statistical decision problem where an agent aims to optimize his decisions while improving his knowledge at the same time [65]. Bandit algorithm is a framework to maximize overall users' engagement by balancing a trade-off between exploitation (show best items) and exploration (show new items) [66]. The exploration vs exploitation dilemma states that we should use exploration to acquire new knowledge about an environment or setting whereas we use exploitation to make decisions or predictions while using existing knowledge [66]. This exploration–exploitation dilemma can be closely related to a RS that needs to emphasize on the existing information to raise user interest while discovering new information to improve user experience. Most of such scenarios are based on user' computational behavior and the environment surrounding him, therefore can be better addressed by CARS. In CARS, we need exploration to make recommendations based on ongoing situation that may sometimes bring some suboptimal results but in the long term, exploitation gathers more rewards to produce long term satisfactory results. Multi-armed bandit (MAB) including ε-greedy, LinUCB, Bernoulli and Thomson Sampling, and the contextual bandit are important bandit algorithms [66] mostly used in recommendation scenarios. In the MAB algorithm for a CARS, an agent chooses an action at each stage and receives a reward from it; and the decision maker decides which arm to use to have maximum total reward. The contextual bandit algorithm, on the other hand, estimates rewards using feature-based prediction model. Bandit algorithms are potentially useful to deal with data sparsity and cold start problem inherent of RS. A typical ε-greedy algorithm chooses pre-defined items based on current knowledge with a probability of 1- ε and randomly chooses other items with a probability of ε [66]. To improve the adaption of ε-greedy, contextualized ε-greedy algorithm had been proposed to compare users' ongoing context with a class of situations and be able to adaptively balance between the exploration and exploitation [67]. Contexts that are not fully observable and are dynamic had been mapped to MAB problem. In such cases, Thompson sampling had been used as the bandit strategy to compute the expected reward for each of the items through its click-through-rate [68,69]. A non-parametric non Bayesian Thompson Sampling had been used to randomly select a bootstrap sample (resampling) to find the maximum likelihood estimation (MLE) of model coefficient and discover the best item [70]. A dynamic context drift model involving particle learning and online inference algorithms had been proposed to capture time varying behaviors of the reward in contextual multi-arm bandit problems in CARS [71]. Contextual bandit algorithm [71] had been extended to introduce user relation information to model dependency among users [72].

As observed, the above-mentioned techniques focus primarily on the prediction as a stepping stone to make good recommendations. **Learning to Rank (LTR)** instead optimizes a model for its quality using a ranking function. LTR refers to supervised machine learning techniques to train a model in the ranking tasks [73]. These techniques are categorized into three: (i) pointwise takes a single document to predict how relevant it is for the current query whereas (ii) pairwise takes a pair of documents and (iii) listwise takes a list of documents to come up with the optimal ordering for it. Recommendation is a ranking problem and LTR is a key element for personalization. A ranked list of contextualized queries for each document were re-ranked by assessing performances of subsets of query terms based on most viable context items in a time-aware recontextualization system [74]. To further improve the effectiveness of event based network, different recommenders were used as features for LTR events through a listwise approach [75]. Factorization machine algorithm [41] making up a general predictor that can work with any real valued feature vector, had been combined with pairwise LTR techniques to address the problem of optimizing a ranked list of items in a CARS [76]. LTR had also been used in conjunction with TF algorithms for optimizing ranking performance [43,44].

Besides these algorithms, some classical ones used in CARS include Google's PageRank algorithm [77], Bayesian networks [78, 79], ontology-based model [80] and Support Vector Machine (SVM) [81]. A general trend for the choice of algorithms in CARS as seen in our chosen publication is shown in Fig. 1.

Fig. 1 shows that MF is the most widely used algorithm to design and develop CARS in all these years. TF inherits most of its properties from MF [38] and is found to be the second most popular choice in CARS after MF. However, TF methods have a large number of parameters to fit in memory and these methods suffer from dimensionality reduction problem [38], which make it a less favorite choice compared to MF methods in CARS. The other algorithms (LDA, Markov, bandit and LTR) are almost at the same level of usage as seen in Fig. 1. For example, Markov models are experimented in context-aware cases when there is a sequence prediction or recommendation problem [57]. LDA has several variants, out of which, the context-aware LDA can be a promising technique to extract rich contextual factors from the data [53]. Bandit algorithms, more specifically the contextual bandit algorithms, can be used to balance between exploration and exploitation dilemma faced by online recommendations [67]. The figure also shows that the bandit algorithms, LDA and Markov methods are slowly getting grip in CARS over these years, although still not as popular as MF or TF. Ranking is naturally posed as a recommendation task that can be implemented through LTR [73]. That is why the LTR algorithms are often used as sideline techniques with other CARS methods to deal with contextual information. The reason for less number of publications in 2017 is that by the time we started this survey, we could only find few publications fulfilling our selection criteria.

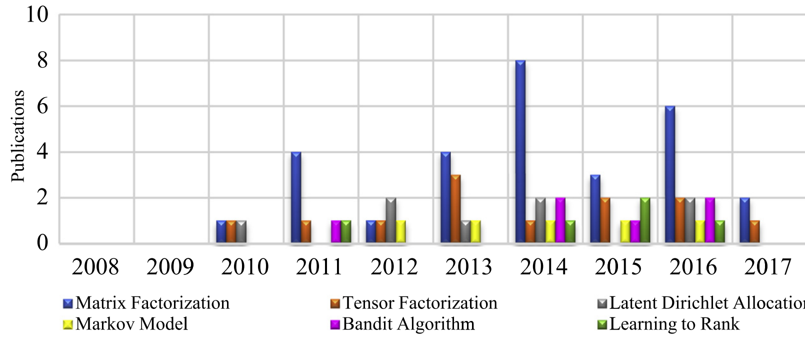

Fig. 1. Classification by algorithms.

4. Dimensionality reduction techniques used in context-aware recommender systems

In traditional RS, most of the sources of the data are represented as large matrices (users–items ratings) which is computationally expensive and inefficient to handle. This has motivated the researchers to find some natural or alternative solution to depict the problem in simpler ways. A dataset with many features naturally leads to high-dimensional space and consequently leads to sparsity with maybe only a small number of features per object. This is known as curse of dimensionality [21]. Dimensionality reduction techniques help to mitigate this problem by converting high dimensionality into lower space. Commonly used dimensionality reduction techniques in the general setting of a RS are : Principal Component Analysis (PCA) consisting of well-known technique for preserving most salient features by a low rank matrix; and Singular Value Decomposition (SVD) including techniques to factorize a rectangular matrix [82]; MF [24]; Self-organizing map as a two layer network model that preserves distances in the outer space [83]; Random Indexing to approximate high dimensional model into a random space of lower dimensionality [84]; Restricted Boltzmann Machines as a stochastic artificial neural network to model tabular data into lower dimensionality [24] and other low rank measures as discussed in the literature [85]. The Netflix competition [24] with a large sparse dataset motivated many researchers to develop model-based algorithms with a generalized stochastic model Restricted Boltzmann Machines.

General component of traditional RS has a typical data record of the form <*user, item, rating*>. In contrast, CARS have the record of the form <*user, item, context, rating*>, where context is an additional dimension and may consists of any number of contexts [7]. In contextual pre-filtering and contextual post-filtering approaches, the multi-dimensional contextual recommendations can be reduced to the standard two-dimensional User $\times$ Item recommendation space through any traditional based dimensionality reduction methods (PCA,SVD or others discussed above) [2]. Contextual pre-filtering was used as a dimensionality reduction technique to prune out extraneous contexts before the MF method was applied to mitigate the cold-start problem [86,87]. Unlike contextual pre-filtering and post-filtering approaches, the contextual modeling provokes to true multidimensional recommendation functions. Therefore, we need predictive, regression, probabilistic or classification-based models, or maybe heuristic calculations to include contextual information in the recommendation process. Multi-dimensional context-aware data consists of context dimension(s) (e.g. time dimension) which in turn consist of context situation(s) (e.g. week as situation) that may have a specific condition(s) (e.g. weekend as condition) [7].

Context is not incorporated as a separate dimension in the basic CAMF [27,28,31,33,37] but as the interaction between the items and different contextual situations (Eq. (1)), with each contextual condition referring to a specific value of contextual factors. Therefore, dimensionality reduction in CAMF (linear with the number of contexts) obtained equal or approximate results through standard MF in a predictive mode [27]. Probabilistic MF method assumes fewer latent factors and factorizes into k latent matrices [29] compared to CAMF with two matrices and more latent variables as the contexts that influenced user rating behaviors; and therefore achieved a low rank MF on the original rating matrix [30,32,36]. Though CAMF is factorized into lower dimensional representation from two entities (users and items), the number of contextual factors in the form of biases may grow exponentially with the number of contexts. One way to limit the contextual features is to select the best contextual conditions from among the various available choices based on the application domain and viability of it nature [30,31,33,37]. Since CSLIM were built on the top of CAMF [27] models, CSLIM methods inherently represented users' and items' vectors in reduced feature space [35].

Tensor is a natural fit to CARS since handling multiple entities in the recommendation algorithms is beyond the capacity of MF. With the introduction of new dimension, naturally comes the data sparsity leading to inconsistencies and failure of the learning algorithm. Various approaches are used in typical tensor-based RS to deal with dimensionality issue. Singular value decomposition (SVD) is a well-known choice for lower-rank tensor decomposition for the cases where there are two dimensions [85]. For tensors of order ≥ 3: Candecomp/Parafac (CP), Tucker decomposition like HOSVD, Tensor Train decomposition or sometimes hybrid formats generalized from matrix SVD and PCA are used as dimensionality reduction techniques [85]. In a simpler case, the tensor had been reconstructed in CubeSVD [38] after applying various smoothing techniques on the content rather than on the tensor. Later, HOSVD had been used to perform latent semantic analysis on the data to discover latent semantic features of the ternary relations in social tagging RS [38]. Candecomp/Parafac decomposition had been used to reduce tensor (time) dimensionality in a link prediction network [38].

Generalizing TF to any dimensional model leads to the implementation of TF methods in CARS. A modified version of stochastic gradient descent (SGD) algorithm from HOSVD decomposition approach had been used to reduce the dimensionality of N-dimensional tensor into factor matrices [40]. PITF reduction [41] had served the purpose of targeting an optimal ranking to reduce high order tensor dimensions [42]. The high dimensionality of TFMAP [44] had been controlled by employing a buffer (limited set of items) that is assigned to each (user, context) pair. In another model, the latent space (users, items and contexts) were represented using a fixed number of dimensions to reduce the overall complexity of the model [43]. The higher number of features through a pointwise preference model to achieve efficient computation in n-dimensional tensors had been reduced to a lower number through pairwise interaction between different entities [45]. Contextual operating tensors (users and items) had been factorized into corresponding contextual operating matrices and each context was represented as a latent vector to reduce dimensionality [46]. Dimensionality was also controlled by limiting the number of entries in a ternary tensor [47].

With the inclusion of contextual factors in the topic-keyword distributions of the original LDA [48] model, the dimensionality of underlying topical embeddings also increases. Topic models are unsupervised techniques and has the inherent ability to analyze large textual document corpus into a low-dimensional representation of documents by automatically discovered topics [50,53]. To finite the k-dimensional feature space, a CARS had applied the concept of closed subspaces or projections from the well-known Gleason's theorem [50]. The dimensionality of users' digital traces into topics were controlled by defining each user profile as the weighted sum of projections from users' digital traces relevant to their interests [54]. The high dimensionality of feature space owing to contextual factors in CARS was also controlled by capturing the association between the features by the latent factors from the LDA training model [51,52]. An analogous approach uses semantic contexts from the ontology to finite the number of contextual features [55]. Standard PCA had been used to reduce the dimensionality of data and to extract latent contexts from the mobile sensors [33].

Markov chains with its capability to incorporate more features leads to high dimensional state space. High dimensionality in Markov process had been reduced through different analytical tools such as PCA [59]; LDA [61]; and sometimes using heuristics to explicitly view k-most recent items [60]. Though Bandit algorithms in CARS had not explicitly addressed dimensionality reduction through their methods, naturally bandits adapt to the changing users' preferences evolving in the space of items that ultimately leads to optimal or close to optimal recommendations [67,69, 70,72]. LTR in CARS had exploited the functionality of PCA [76]; LDA [74] and Bayesian Personalized Ranking to leverage personalized contextual signals in the recommendation problem [75]. We have drawn a distribution of our chosen publications with respect to whether they have used dimensionality reduction techniques evolving over these years to see the trend as shown in Fig. 2.

As can be seen in Fig. 2, MF is the most popular technique for dimensionality reduction in CARS. This is because MF is inherently based on a low rank approximation of feature space [25]. LDA can be seen as the second most popular technique to reduce dimensions in CARS as it has the natural tendency to represent documents as low-dimensional representations. From the mathematical perspective, LDA is a dimensionality reduction technique since it reduces the feature space of dimensions from the vocabulary size $|V|$ into some k number of topics specified by the user [48]. As we know, TF is one of the promising techniques to incorporate contextual information in CARS. However, it is quite evident in the literature that the biggest dilemma in TF is the curse of dimensionality [38]. Thus, it is challenging for the researchers and designers to reduce the large number of dimensions in TF methods within CARS, where each context maps to a new dimension. This requires lot of research, knowledge from factorization methods (as in MF) and may need to trade-off between two competing aims: increasing number of tensors with the context features and loss of information due to dimensionality reduction. The other techniques (bandit, Markov, LTR) are found to be almost at the same level of usage in CARS to reduce dimensionality. The findings from our collection of publications revealed to us that these techniques have employed dimensionality reduction methods from well-known techniques such as PCA or LDA to address this dilemma in CARS. This implies that bandit, Markov and LTR techniques are still under theoretical research for the subject of dimensionality reduction and need much more investigation. Fig. 2 shows less number of publications and algorithmic techniques in 2017. This is because there were fewer publications by that time that fulfilled our selection criteria.

5. User modeling techniques used in context-aware recommender systems

A traditional RS can only make high-quality recommendations to the users once they model their preferences in the recommendation process [88]. A user model consists of knowledge about the user that is encoded either explicitly or implicitly and is used by a RS to improve users' interactions with the system [89]. Context plays an important role in making recommendations since users' behaviors are greatly affected by their current situation(s). The user modeling techniques used in traditional RS lacks the contextual information that is why developers and researchers in CARS are seen to design and develop contemporary solutions for user modeling [88]. User experience had been modeled in the general setting of RS in the form of users' satisfaction, attitude, behavior and interactions, but these systems fail to explain how and why the user experience within the RS has originated. A generic user model for CARS had been proposed lately that stores pre-defined information categories for the user, item, activity and context attributes that are readily usable and adaptable [88]. However, due to the variety of the contexts, there are no standardized user models found in CARS. We have reviewed our chosen publications to see how different authors have explored the subject in their recommendation approaches.

A user states his preferences either explicitly (rating, comments, reviews) or implicitly (through observations) for an item. Contexts in ubiquitous computing can be identified into two views: representational and interactional [22]. In the representational view to context, context is represented as an explicit set of attributes that is identifiable, observable and known in prior [7]. Such a context can be easily captured from a CARS application and does not change significantly over the time [7]. In the interactional view to context, the contexts are not fully observable or are partly observed and are fed into the RS from the data, environment or by analyzing the ongoing interaction of a user/item with the system. Whether using representational view to context or the interaction view, the contexts are incorporated into the recommendation process through contextual modeling to CARS [7]. We have classified our chosen publications to find out how the information is being modeled in the recommendation process and which view (interactional or representational) to context has been adopted.

In the **representational view to contexts** in CAMF methods, representations can be seen as a set of pre-defined number of contexts that were incorporated as the interaction of different contextual factors and ratings in baseline CAMF methods [27,28, 33,75]; social contexts in the form of latent factors of users and ratings in probabilistic MF methods [30,31]; aggregation of users' ratings under varying number of contextual situations with different users/items biases in baseline CAMF models [35]; projection of multiple latent factors from user and item entities as context-specific and intrinsic preferences in CAMF [36]; and weighted product of user latent factors and check-in contextual factors into probabilistic MF model [32]. With respect to representational view to context in TF methods, representations can be seen as the contextual conditions in the form of gender, companion, days of the week and levels of hunger that had been incorporated as the features affecting the rating into a HOSVD-decomposition model [40]; common semantic effects of being virtual (or real), hunger and companionship represented as bias parameters in contextual operating matrices [46]; and geo-topical-social contexts integrated as regularization terms [47] to improve the functionality of TF methods. In one of the LDA models, representations can be seen as the topic distribution of the documents from the title, abstract and surrounding text to make citations [50]. Query terms and a temporal expression were represented as the information to retrieve the top-k time units using LTR algorithms [74].

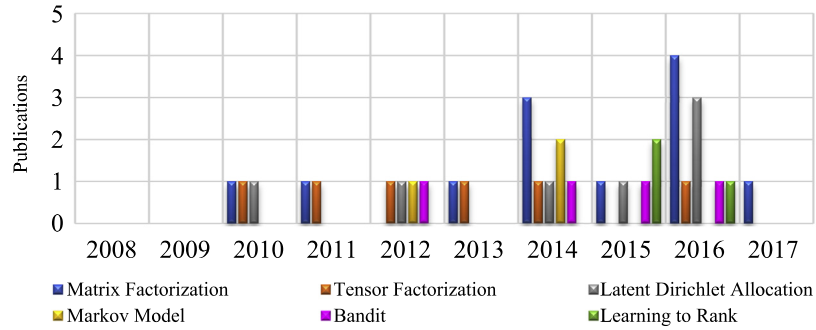

Fig. 2. Distribution of the use of dimensionality reduction techniques over the years.

The **interactional view to context** can be seen in the CAMF methods as partly observable implicit preferences of students integrated as parameters in the basic CAMF method [28]; RSVPs information of the users in a probabilistic MF model [37]; location and time contexts from the sensors into CAMF model [33] to represent latent factors of the ratings. Similarly in the TF methods, interactional view to context had also been used to derive time, sequential and location information from the transactional data into a PARAFAC decomposition model [42]; motion and location with different contextual situations from the GPS tracer to represent underlying interactions through CP model [44]; timestamp as context specific latent vector into CAMF method [46]; and as season and sequence features into Alternating Least Square (ALS) optimization learning [45].

Another set of interactional view to context can be seen in context-aware LDA models: in making common context-aware preferences based on temporal, geographical and system information through LDA [51,55]; building unified user profiles through partly observed users' traces (posts, tweets etc.) [52,54]; comparing topic diversity through topic distribution between the documents and context [74]; and through a knowledge base [55]. Sequence of timestamp events [61,63], song patterns [59] and clicks on web pages [60] resulting from the interaction of the logs are modeled as states in Markov processes to leverage contextual changes and adapting to varying users' interests. Interactive view to context can also be seen as reward in terms of user clicks [67, 69,70] in contextual bandit algorithm; implicit user feedback [68] and user clicks [71] in multi-arm bandit problems to influence personalized users' preferences in real time. Contextual signals in the form of groups, distances and content were inferred to include users' preferences in the scoring function of LTR [75]; and as implicit feedback scenarios in the combination of ranking based factorization machine [41] with pairwise LTR techniques [76] to make top-N recommendations. We have drawn a distribution of our chosen publications with respect to the view to context over the years used in different algorithms as shown in Figs. 3.a and 3.b.

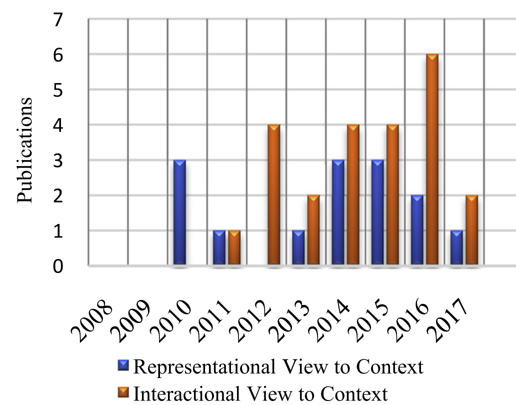

Fig. 3.a. Distribution of context views.

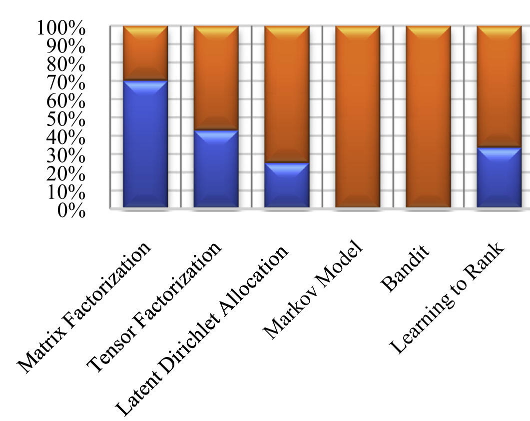

Fig. 3.b. Distribution of algorithms over the context views.

"It can be seen in Fig. 3.a that the interactional view to context is a well-known way to incorporate the context into the recommendation process. As there is lot of hidden information in the data and user behavior [7] that can be implicitly captured from within the context-aware applications, it makes the interactional view to context a favorable choice to model contextual information in CARS. Representational view to context can also be used to model contextual information in cases when we have some priori knowledge about contextual attributes within the recommendation process.

As can be seen in Fig. 3.b, interaction view to context is used by all algorithms in CARS for the obvious reason that these algorithms try to capture contexts from the latent information. This figure also shows that CF based algorithms (MF, TF and in some cases hybridization with LDA or LTR) with sufficient information about users' behavior typically make use of representational view to context. However, in some papers these algorithms take implicit information from the users' behavior too to address data sparsity problem. Bandit algorithms and Markov algorithms, though not as popular as CF algorithms, are mostly based on interactional view to context and try to capture implicit information from latent users' preferences and data. That is why these algorithms (bandit and Markov) are well-suited and employed for the environments like news domain where there are more or less the constant number of items and almost all items have little past feedback as articles expire too soon.

6. Evaluation metrics and datasets in context-aware recommender systems

Datasets

In the era of Big Data, RS is one of the most sought-after research topics applicable to both academia and real-life applications. The benchmark datasets[1,2] for the traditional RS categorized according to the application domains are as follows: for movie recommendation: Movielens, Yahoo!, Netflix prize, MovieTweetings; for music recommendation: Last.fm, Yahoo music, Amazon audio and Audioscrobbler; for mobile recommendations: Mobile App Retrieval, frappe; for books: Book-crossing, Amazon books dataset; for food: Chicago Entree, Yelp; for merchandise : Amazon; for health: Hospital Ratings, Texas healthcare information, NHS diabetes, mental health, cancer and joint registry datasets; and for scholarly articles: CiteULike, Bibsonomy, CiteSeer, ACM, DBLP.

Likewise, CARS are applicable in both research and real-time applications on different datasets; and application domains such as event-based social networks [75], mobile applications [14], e-learning [12], tagging system [38]. Each of such systems has a different objective, different audience and therefore leads to different requirements for the RS being used. As discussed by Adomavicius and Tuzhilin [7], the application domains for CARS are limited. This is particularly true since there is a lack of publicly available datasets for CARS [90]. Big companies such as Facebook, Google, Youtube and Bing invest heavily on recommendation services, but they have released little or no information about their implementation of recommendation services; therefore, the research community in RS cannot fully benefit from them.

There are three kinds of datasets in the field of CARS: real-world benchmark, semi-synthetic and fully synthetic datasets. Much of the experimentation and evaluation in CARS is accredited to the benchmark datasets that are used in the general setting of RS. But these public datasets contain very limited context-aware ratings and are not adequate to evaluate CARS algorithms very well. One possible solution is to crawl such data from the well-known websites supporting each distinct application domain. But this is a very time-consuming process and complicated due to involuntary response from the companies or users, copyright issues or privacy reasons. Another practice that is lately seen in CARS research is to enable researchers and application developers to create their own synthetic or semi-synthetic datasets for CARS by simulating the contextual attributes that they want to evaluate their algorithms on. Besides these options, there are very few publicly available CARS datasets. For instance, CARSKit[3] [91] is an open source Java-based CARS engine that implements algorithms and provides datasets and evaluation metrics specific to CARS which are freely usable, modifiable and distributable under GNU General Public License.[4] However, the number of datasets available through CARSKit is very limited. We have classified our chosen publications according to the application domains and the datasets being used.

Yahoo webscope movies and MovieAT are two real-world datasets that had been regenerated to create semi-artificial datasets in baseline CAMF methods [27] and for implementing TF methods [40]. A contextual movie dataset consisting of time, place and companion as the contextual dimensions with coarse granularity for each dimension had been created initially for a movie RS [92] to be used in different CARS approaches [27,40,46]. Contextual movie datasets were also created from Moviepilot and Filmtipset datasets [86,93] by incorporating potential pieces of contextual information into the data. Benchmark dataset Movielens was synthetically generated to add contexts in the form of temporal [46] and topical information [55]. Lastfm dataset was regenerated by including timestamped features [63,72,76]. Similarly, LastFM, Movielens, Netflix, VOD and grocery store datasets were synthetized to include sequence and season as the contextual information [45]. Yahoo music and Yelp datasets were also regenerated to include users' ratings under certain contexts to address music recommendations [76]. A product review dataset was created by crawling a product review site Ciao with authors, raters and their relations as the contextual information [30]. Permission to conduct research on Coursera MOOC datasets with specific contextual features was obtained for the question recommendation problem [28].

Food dataset [78] had been used for evaluating top-N recommendation in CARS [35,43] and for capturing semantic relationship between contexts and the entities [46]; and Restaurant dataset [94] had been created exclusively for CARS through user studies [95]. Users' assessment with the contextual effects on music preferences had been captured through web applications to make InCarMusic dataset [96] to be used for top-N recommendations [35]; inside MF [86] and TF methods [40] in CARS. Another contextual music dataset was created from Appazaar (http://appazaar.net/) application for top-n recommendation problem [44]. Traditional RS datasets Playlists for songs and CiteULike for articles tagged with contextual features had been used for a query driven CARS [52].

Social networking contextual datasets were created by crawling Renren (www.renren.com) and Tencent Weibo (t.qq.com) websites [31] and from Glue social network [42] through incorporating personal and interpersonal contextual factors for social recommendation problem. Widely used Meetup dataset for general RS was used in social based event CARS [37,54,75]. Tuent (https://www.tuenti.com/), a social network website had been crawled for capturing user-video clicks to be used in making top-N recommendations [43]. Medium.com used widely in typical RS had been used for social event CARS problem [54].

Point of interests and related contextual features were extracted through an application with the help of users who participated in the experiment [32,33,86,97,98]. Mobile application usage datasets were created with contextual information collected from Frappe application [43]; and through user studies [51,61,67]. Twitter dataset tagged with geo-based features [47] and digital traces of the users [54] had been synthesized for CARS. CiteSeerX had been crawled to make a CARS dataset for a citation based system [50]. A context aware click log dataset had been created by crawling an online fashion retailer site by adding demographics and personal contextual features to it [60]. Yahoo user-rating and a web navigational dataset had been synthesized for different contextual features for an interactive RS [68].

CARS for news mostly use typical RS based datasets to produce semi-artificial contextual datasets. Yahoo! Today Module dataset [69–72]; KDD Cup 2012 Online Advertising [69,70,98]; New York Times Annotated Corpus and Wikipedia dumps [74] were regenerated to populate them with contextual features to get click streams in news recommendation problem. Although we see some development in datasets specific to CARS, at the same time it can be seen that these synthetized datasets are usually not publicly available. There are some CARS datasets available in CARSKit[5] under the terms of the GNU General Public License such as InCarMusic [96], Foods [78], Resturant [81], Frappe [99], TripAdvisor [100] and Adoms [7] that were also cited in many CARS publications.

1 https://gist.github.com/entaroadun/1653794.

2 https://github.com/ArthurFortes/Datasets-for-Recommneder-Systems/.

3 https://github.com/irecsys/CARSKit.

4 http://www.gnu.org/licenses/.

5 https://github.com/irecsys/CARSKit.

Table 1
Datasets used in general RS and CARS.

General	MovieLens, Netflix, Meetup, Grocery, LastFM, VOD, Medium.com, Twitter, Wikipedia dumps, facebook, Playlists, CiteULike, Yahoo Yelp, New York Times, KDD Cup 2012, Delicios
Contextualized	Tourism, InCarMusic, Appazaar.net, Product review, Foursquare, Foods, Resturant, Tencent Weibo.com, Renren.com, MOOC, Adom (movies), Frappe, getGlue, CiteSeerX, mobile logs, Zalando click logs, TripAdvisor, Adom, Frappe

In Table 1, we have shown datasets (general and context-specific) used in CARS.

As can be seen in Table 1, the datasets for general RS are mostly open source and are publicly available. However, the contextualized datasets are mostly proprietary (privately owned) and are not usually released to the general public. They are usually generated for the domain for which a CARS is to be developed like POIs based, mobile based or such. This becomes challenging if we want to learn and implement recommendation algorithms for CARS.

Evaluation metrics

We need to use different evaluation approaches to test the effectiveness of a RS. The goal of any RS is to trigger a user to interact with at least one of the items shown in the recommendation list. We can test the recommendation results in real-time on real users to see whether they can help users find relevant items from the available recommendations and therefore result in overall better user experience. Such an evaluation is called online evaluation [101]. There are variety of online evaluation metrics such as impressions, click-through-rate (CTR), conversions and others [102]. However, online evaluation is not always a feasible approach for different reasons. For example, it is time consuming to test all possible RS algorithms under certain configurations and constraints; and users may be negatively affected by an algorithm that might be suitable for one use-case but not for all scenarios [103]. Therefore, we make use of offline evaluation in such cases and do not change the recommendation in the running system [103].

Evaluation metrics in the general setting of a RS can be categorized broadly as metrics used to evaluate the rating predictions made by a RS such as Mean Absolute Error (MAE) and the root-mean-square error (RSME); and metrics used to evaluate top-N recommendations such as Precision, Recall, Mean average precision (MAP), Area Under the ROC Curve (AUC) and Normalized Discounted Cumulative Gain (NDCG) [101,103] and their variants. One of the most effective evaluation techniques come through users' studies that measure users' satisfaction through explicit ratings [95]. Although we could not find many evaluation approaches that are specific to CARS, we found out the coverage of traditional RS evaluation methods used for CARS.

Traditional mean absolute error (MAE) in a RS is the average of all absolute ratings [103] as shown in Eq. (5).

$$MAE = \frac{1}{\mathcal{R}} \sum_{\tilde{r}_{ui}}^{\mathcal{R}} |\tilde{r}_{ui} - r_{ui}| \tag{5}$$

where $\mathcal{R}$ is the number of the items rated by user, $\tilde{r}_{ui}$ is the predicted rating of a user u and item i; and r_{ui} is the true user rating. Similarly, RMSE is the square root of mean of squared differences between predicted and actual ratings. To measure the accuracy of rating predictions, traditional MAE had been computed to compare CAMF and basic TF methods in a usual way [27]. Traditional IR metrics for error measures and for evaluating lists of recommendations do not fit to the evaluation of CARS since we need to add contexts as additional constraints. The true ratings r_{ui} in traditional MAE (Eq. (5)) had been replaced to include context(s) c in the form of r_{uic} to accommodate: interpersonal influence matrix besides the latent user and item feature matrices [31]; context as a separate entity [40], as context aware representations [43] and as context-specific and intrinsic latent vectors [46]. Users' interest, geographical, social and categorical relevance scores were added in the MAE and RMSE metrics to match these relevance scores into the preference score for predicting the performance of a point-of-interest model [33]. Context as an event had also been incorporated as the additional parameter in the basic RMSE measure to evaluate the test data in the social recommendation problem [37]. The actual ratings in RMSE had been replaced by likes or check-in tags between users, items and different POI contextual conditions [33]; estimated helpfulness ratings between raters and reviews [30]; hunger and different levels of virtual or real as contexts between users and items [43]; latent vectors between users, items and contextual factors [46] and multi-contexts under a single rating [36].

Compared to the traditional metrics to provide a list of top-N recommended items for each user, context had been included as an additional constraint to provide top-N items using precision, recall, MAP or sometimes F1-score for each <*user, context*> pair [32,35–37,45,47,51,51,54,63,74,76]. Traditional MAP was presented as an optimized smoothed version of MAP where the average precision was replaced by the average of binary relevance between users/items under certain contexts computed through gradient descent methods of TF techniques [43,44].

Traditional Hit@k with hit as a positively rated POI by the user had been used to measure the percentage of hits among the top-K recommendations [33]. Similarly, NDCG had been applied to compute average position indices among top-k recommendations [33,50,75,76]. Standard machine learning accuracy measures : AUC and the HitRatio had been extended to measure every context information for evaluating the ranking performance in social networks [42]. HitRatio had also been used to compute the probability of the songs [52]; and probability of movies [55]. AUC had been used as a binary classification metric for a ranking predictor [76]. Mean Reciprocal Rank (MRR) was used to evaluate the performance of CARS by returning a ranked list of answers to queries with different contexts [76]

CTR is most widely used online evaluation metric that is the average number of the clicks by recommendation (Wu et al., 2016). CTR was defined as the reward of a document when a user selects a document in contextual bandit algorithms [67,72]. CTR had been computed as the total number of hits at each step over total number of recommendations in multi-arm bandit problem [68,71].

A few metrics specific to CARS evaluation as found in the literature include density metrics to find the performance of top-N context-aware recommendation by measuring the levels of sparsity for multiple ratings in contexts [35] and co-citation probability metric to measure the probability of two items by their popularity in the past [50]. Users' feedback is usually evaluated as the utility for the recommendation item. Average utility had been used as the evaluation metric to evaluate bandit recommenders over different sessions in interactive mode [68] and to find the change points in such recommenders [68]. Precision from the bandit algorithms was computed on the ranked list of items in an item-based recommendation problem [72]. Cumulated normalized reward had been used an evaluation metric to measure the effectiveness of collaboration and user-based analysis [72]. Besides the overall CTR (average reward of a trial) for online evaluation, replayer method that utilizes historical logs and weighted vector of each advertisement was used as offline evaluation method in bandit recommenders [70]. Popularity bias was used as an evaluation metric to measure the diversity of different RS [63]. Similar to item diversity [103], matrix diversity had been defined as the mean RMSE of all slice pair of tensors to measure the performance of a CARS. Average rank is a metric for ranking evaluation which had been used to measure the performance of sequential nature of the data through Markov methods in a RS [60]. Yang et al., [28] devised their own evaluation metrics to measure on average how many students/questions are recommended to a question/student to show the load balance statistics of both entities. Besides that, they also introduced an overall community benefit metric as the aggregated relevance scores to measure the performance of the system. In Table 2, we have shown evaluation metrics used in CARS.

As can be seen in Table 2, evaluation metrics for CARS are not as widely used as those for the general RS. This is because, these evaluation measures were specifically designed to evaluate the ongoing recommendation problem in hand. If some researcher wants to re-apply that evaluation measure for another context-aware problem, it requires modification according to the context(s) that are to be used for that particular application.

7. Discussion and future research directions

CARS are making a huge impact in the research and development industry as they play a major role in predicting and anticipating users' needs in response to their behavior for real-world scenarios. With the emergence of context in RS, many researchers start improving the classical recommendation algorithms by incorporating various kinds of contexts in the recommendation process. This has resulted in considerable paradigm shift in the field of RS from being generic towards context-awareness [7–9]. Nevertheless, the identification of the correct algorithm for the type of application domain for which RS is to be built remains a challenge for the application developers [16,19]. The main goal of this review is to collect the information about how the algorithms used in CARS differ from those used in traditional RS. This survey also shows an evolution of these algorithms used in CARS over the time. Besides, it also emphasizes how to incorporate relevant contexts from various sources into a recommendation model that requires formulating the correct performance measures to test such use cases.

The state-of-the-art algorithms in CARS are mostly derived from the traditional recommendation algorithms. They are evolving over time and can be implemented across different application domains. However, it could remarkably ease the task of context-aware recommendations if there is a single standardized framework in CARS that is flexible enough to accommodate diversified contexts and application domains. In this way, the research community can contribute to improve that single framework instead of proposing different variants to fit to each specific application domain.

Curse of dimensionality is an inherent problem in CARS that is preventing the research community to work significantly in the field. The well-known dimensionality reduction techniques used in CARS include factorization methods, PCA, LSI and the more generative LDA model that originally stem from SVD with slight transformations in minimizing least-squared error versus input [82,85]. Besides these techniques, we also find the use of heuristics to cope with the increasing number of dimensions in CARS. However, the factorization methods (MF, TF or probabilistic MF) received greater exposures compared to similar techniques on the same datasets mainly for latent variable decomposition. Dimensionality reduction comes with a cost therefore there should be a justified balance between the lesser number of dimensions and retaining the necessary information.

The CARS applications developed to handle the representational view to context in learning users' profiles [8,92] neglect the dynamicity of contexts. Although there are some algorithms like bandit and Markov process that show fuller tendency towards interactional view to context, but they are rarely used in pure CARS application. User models built on partial users' and contextual information result in poor performance of CARS. This implies the need of investigating possible mechanisms to identify contextual changes and adjust them in the recommendation models during runtime. To deal with the context management research, it is very crucial to gather heterogeneous contexts from the environment and support reasoning rules to infer context facts from the raw data. This requires building a pre-implemented user modeling framework for CARS [88] that can serve as a reference framework for creating more specialized models.

The crux of CARS algorithms lies in contextual modeling that directly incorporates context in the recommendation model [7]. Exhaustive contextual modeling had been identified as a major issue in CARS, which suggests that relevant context must be identified before applying it [19]. This is a challenge which involves investigating standardized methods to incorporate contexts in the recommendation processes.

There seems to be an urgent need for building datasets specific to CARS. Although there are a few besides the benchmark datasets used in the general setting of RS, most of these CARS datasets are not publicly available perhaps due to the privacy and copyright issues. Providing academics with proper datasets can move the CARS area towards real-world applications. Another common practice seen in CARS is that they are generally testing the effectiveness of their applications using traditional offline evaluation mechanism. Such measures may provide handy results, but they might later contradict the results obtained in online settings. Therefore, developing proper evaluation strategies is very crucial to advance the state-of-the-art in CARS.

We have found from the study of these publications that contexts are mostly coded into the recommendation process without any formal investigation. Emotional states of the users like hunger, companionship were considered in the contextual modeling [40, 46] but there are very limited examples of such work. Similarly, there are very few examples [55] that use knowledge bases to infer richer contexts. Integrating other technologies such as sentiment analysis, ontology, clustering, or association rule mining into the CARS research can greatly enhance context-aware recommendations for practical scenarios.

8. Conclusion

This paper presents the literature review on CARS processes and methods based on the papers published between 2008 and 2017. The study was conducted with the goal to help the research community in CARS understand how context can be incorporated in typical recommendation algorithms. The main focuses of our paper are on the integration of context in the recommendation methods; reducing the dimensionality of numerous contexts; tailoring observed, partly observed or unobserved contexts to learn user profiles; and reviewing the evaluation strategies and benchmark datasets to measure the strengths and weakness of these studies. Despite the comprehensiveness in the state-of-the-art CARS, there is still no generalization in the algorithmic approaches being used in CARS. This could be partly because of the unexplored knowledge of contextual preferences, nature of the data exploited in these studies and lack of objective validation methods. Besides highlighting the gaps in the literature, this survey also presents several research challenges in the form of opportunities for the researchers and application developers to further investigate. These challenges are particularly related to (i) build a generalized CARS framework (ii) develop standardized context-aware recommendation approaches (iii) include context identification and incorporation

Table 2
Evaluation measures used in general RS and CARS.

General	MAE, RMSE, Precision, Recall, MAP, F1-score, NDCG, HitRatio, AUC, CTR, MRR
Context-aware	Average utility, Cumulated normalized reward, Replayer, Popularity bias, Average rank, Co-citation probability, Overall community benefit, Matrix diversity

(iv) dimensionality reduction techniques (v) standardized user modeling (vi) benchmark datasets and evaluation mechanisms. In addition to these challenges, CARS also need to address the inherent issues for a RS such as cold-start, sparsity, self-biasness, runtime mechanism as well as privacy and copyright issues.

Acknowledgments

I would like to express my very great appreciation to Chen Ding for her valuable and constructive suggestions during the writing of this survey. Her willingness to give her time so generously has been very much appreciated.

References

[1] P. Zikopoulos, C. Eaton, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, McGraw-Hill Osborne Media, 2011.

[2] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (2005) 734–749, http://dx.doi.org/10.1109/TKDE.2005.99.

[3] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: A survey, Decis. Support Syst. 74 (2015) 12–32, http://dx.doi.org/10.1016/j.dss.2015.03.008.

[4] D. Goldberg, D. Nichols, B.M. Oki, D. Terry, Using collaborative filtering to weave an information tapestry, Commun. ACM 35 (1992) 61–70, http://dx.doi.org/10.1145/138859.138867.

[5] A.K. Dey, Understanding and using context, Pers. Ubiquitous Comput. 5 (2001) 4–7, http://dx.doi.org/10.1007/s007790170019.

[6] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, in: Proceedings of the 2008 ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2008, pp. 335–336, http://dx.doi.org/10.1145/1454008.1454068.

[7] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, in: Recommender Systems Handbook, Springer, Boston, MA, 2015, pp. 191–226, http://dx.doi.org/10.1007/978-1-4899-7637-6_6.

[8] K. Haruna, M. Akmar Ismail, S. Suhendroyono, D. Damiasih, A.C. Pierewan, H. Chiroma, T. Herawan, Context-aware recommender system: a review of recent developmental process and future research direction, Appl. Sci. 7 (2017) 1211, http://dx.doi.org/10.3390/app7121211.

[9] N.M. Villegas, C. Sánchez, J. Díaz-Cely, G. Tamura, Characterizing context-aware recommender systems: A systematic literature review, Knowl.-Based Syst. 140 (2018) 173–200, http://dx.doi.org/10.1016/j.knosys.2017.11.003.

[10] J. Hong, E. Suh, S.-J. Kim, Context-aware systems: A literature review and classification, Expert Syst. Appl. 36 (2009) 8509–8522, http://dx.doi.org/10.1016/j.eswa.2008.10.071.

[11] H.-L. Truong, S. Dustdar, A survey on context-aware web service systems, Int. J. Web Inf. Syst. 5 (2009) 5–31, http://dx.doi.org/10.1108/17440080910947295.

[12] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, E. Duval, Context-aware recommender systems for learning: A survey and future challenges, IEEE Trans. Learn. Technol. 5 (2012) 318–335, http://dx.doi.org/10.1109/TLT.2012.11.

[13] M. Kaminskas, F. Ricci, Contextual music information retrieval and recommendation: State of the art and challenges, Comput. Sci. Rev. 6 (2012) 89–119, http://dx.doi.org/10.1016/j.cosrev.2012.04.002.

[14] Q. Liu, H. Ma, E. Chen, H. Xiong, A survey of context-aware mobile recommendations, Int. J. Info. Tech. Dec. Mak. 12 (2013) 139–172, http://dx.doi.org/10.1142/S0219622013500077.

[15] U. Panniello, A. Tuzhilin, M. Gorgoglione, Comparing context-aware recommender systems in terms of accuracy and diversity, User Model. User-Adapt. Interact. 24 (2014) 35–65, http://dx.doi.org/10.1007/s11257-012-9135-y.

[16] Z.D. Champiri, S.R. Shahamiri, S.S.B. Salim, A systematic review of scholar context-aware recommender systems, Expert Syst. Appl. 42 (2015) 1743–1758, http://dx.doi.org/10.1016/j.eswa.2014.09.017.

[17] M. Bambia, R. Faiz, M. Boughanem, Context-awareness and viewer behavior prediction in social-TV recommender systems: Survey and challenges, in: New Trends in Databases and Information Systems, Springer, Cham, 2015, pp. 52–59, http://dx.doi.org/10.1007/978-3-319-23201-0_7.

[18] A. Abbas, L. Zhang, S.U. Khan, A survey on context-aware recommender systems based on computational intelligence techniques, Computing 97 (2015) 667–690, http://dx.doi.org/10.1007/s00607-015-0448-7.

[19] I. Ben Sassi, S. Mellouli, S. Ben Yahia, Context-aware recommender systems in mobile environment: On the road of future research, Inf. Syst. 72 (2017) 27–61, http://dx.doi.org/10.1016/j.is.2017.09.001.

[20] R. Kalaivani, R. Sivakumar, A Survey on Context-Aware Ubiquitous Learning Systems, 2017, 14.

[21] E. Keogh, A. Mueen, Curse of dimensionality, in: Encyclopedia of Machine Learning and Data Mining, Springer, Boston, MA, 2017, pp. 314–315, http://dx.doi.org/10.1007/978-1-4899-7687-1_192.

[22] P. Dourish, What we talk about when we talk about context, Pers. Ubiquitous Comput. 8 (2004) 19–30, http://dx.doi.org/10.1007/s00779-003-0253-8.

[23] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, 401 (1999) 4.

[24] G. Takács, I. Pilászy, B. Németh, D. Tikk, Matrix factorization and neighbor based algorithms for the netflix prize problem, in: Proceedings of the 2008 ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2008, pp. 267–274, http://dx.doi.org/10.1145/1454008.1454049.

[25] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (2009) 30–37, http://dx.doi.org/10.1109/MC.2009.263.

[26] B. Mobasher, Context-Aware User Modeling for Recommendation, 2013.

[27] L. Baltrunas, B. Ludwig, F. Ricci, Matrix factorization techniques for context aware recommendation, in: Proceedings of the Fifth ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2011, pp. 301–304, http://dx.doi.org/10.1145/2043932.2043988.

[28] D. Yang, D. Adamson, C.P. Rosé, Question recommendation with constraints for massive open online courses, in: Proceedings of the 8th ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2014, pp. 49–56, http://dx.doi.org/10.1145/2645710.2645748.

[29] R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, in: Proceedings of the 20th International Conference on Neural Information Processing Systems, Curran Associates Inc., USA, 2007, pp. 1257–1264, http://dl.acm.org/citation.cfm?id=2981562.2981720 (accessed June 8, 2018).

[30] J. Tang, H. Gao, X. Hu, H. Liu, Context-aware review helpfulness rating prediction, in: Proceedings of the 7th ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2013, pp. 1–8, http://dx.doi.org/10.1145/2507157.2507183.

[31] M. Jiang, P. Cui, F. Wang, W. Zhu, S. Yang, Scalable recommendation with social contextual information, IEEE Trans. Knowl. Data Eng. 26 (2014) 2789–2802, http://dx.doi.org/10.1109/TKDE.2014.2300487.

[32] X. Ren, M. Song, H.E.J. Song, Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation, Neurocomputing 241 (2017) 38–55, http://dx.doi.org/10.1016/j.neucom.2017.02.005.

[33] M. Unger, A. Bar, B. Shapira, L. Rokach, Towards latent context-aware recommendation systems, Knowl.-Based Syst. 104 (2016) 165–178, http://dx.doi.org/10.1016/j.knosys.2016.04.020.

[34] X. Ning, G. Karypis, SLIM: Sparse linear methods for top-N recommender systems, in: 2011 IEEE 11th International Conference on Data Mining, 2011, pp. 497–506, http://dx.doi.org/10.1109/ICDM.2011.134.

[35] Y. Zheng, B. Mobasher, R. Burke, CSLIM: Contextual SLIM recommendation algorithms, in: Proceedings of the 8th ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2014, pp. 301–304, http://dx.doi.org/10.1145/2645710.2645756.

[36] T. Man, H. Shen, J. Huang, X. Cheng, Context-adaptive matrix factorization for multi-context recommendation, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, New York, NY, USA, 2015, pp. 901–910, http://dx.doi.org/10.1145/2806416.2806503.

[37] Y. Gu, J. Song, W. Liu, L. Zou, Y. Yao, Context Aware Matrix Factorization for Event Recommendation in Event-Based Social Networks, in: 2016, pp. 248–255. http://dx.doi.org/10.1109/WI.2016.0043.

[38] E. Frolov, I. Oseledets, Tensor methods and recommender systems, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 7 (2017) e1201, http://dx.doi.org/10.1002/widm.1201.

[39] T. Kolda, B. Bader, Tensor decompositions and applications, SIAM Rev. 51 (2009) 455–500, http://dx.doi.org/10.1137/07070111X.

[40] A. Karatzoglou, X. Amatriain, L. Baltrunas, N. Oliver, Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering, in: Proceedings of the Fourth ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2010, pp. 79–86, http://dx.doi.org/10.1145/1864708.1864727.

[41] S. Rendle, L. Schmidt-Thieme, Pairwise interaction tensor factorization for personalized tag recommendation, in: Proceedings of the Third ACM International Conference on Web Search and Data Mining, ACM, New York, NY, USA, 2010, pp. 81–90, http://dx.doi.org/10.1145/1718487.1718498.

[42] H. Wermser, A. Rettinger, V. Tresp, Modeling and learning context-aware recommendation scenarios using tensor decomposition, in: 2011 International Conference on Advances in Social Networks Analysis and Mining, 2011, pp. 137–144, http://dx.doi.org/10.1109/ASONAM.2011.56.

[43] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, CARS2: Learning context-aware representations for context-aware recommendations, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, ACM, New York, NY, USA, 2014, pp. 291–300, http://dx.doi.org/10.1145/2661829.2662070.

[44] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, N. Oliver, TFMAP: Optimizing MAP for Top-n context-aware recommendation, in: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, 2012, pp. 155–164, http://dx.doi.org/10.1145/2348283.2348308.

[45] B. Hidasi, D. Tikk, General factorization framework for context-aware recommendations, Data Min. Knowl. Discov. 30 (2016) 342–371, http://dx.doi.org/10.1007/s10618-015-0417-y.

[46] Q. Liu, S. Wu, L. Wang, COT: Contextual operating tensor for context-aware recommender systems, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI Press, Austin, Texas, 2015, pp. 203–209, http://dl.acm.org/citation.cfm?id=28870072887036 (accessed April 29, 2018).

[47] H. Ge, J. Caverlee, H. Lu, TAPER: A Contextual Tensor-Based Approach for Personalized Expert Recommendation, ACM Press, 2016, pp. 261–268, http://dx.doi.org/10.1145/2959100.2959151.

[48] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.

[49] K. Stevens, P. Kegelmeyer, D. Andrzejewski, D. Buttler, Exploring topic coherence over many models and many topics, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, Jeju Island, Korea, 2012, pp. 952–961, http://www.aclweb.org/anthology/D12-1087 (accessed June 11, 2018).

[50] Q. He, J. Pei, D. Kifer, P. Mitra, L. Giles, Context-aware citation recommendation, in: Proceedings of the 19th International Conference on World Wide Web, ACM, New York, NY, USA, 2010, pp. 421–430, http://dx.doi.org/10.1145/1772690.1772734.

[51] K. Yu, B. Zhang, H. Zhu, H. Cao, J. Tian, Towards personalized context-aware recommendation by mining context logs through topic models, in: Advances in Knowledge Discovery and Data Mining, Springer, Berlin, Heidelberg, 2012, pp. 431–443, http://dx.doi.org/10.1007/978-3-642-30217-6_36.

[52] N. Hariri, B. Mobasher, R. Burke, Query-driven context aware recommendation, in: Proceedings of the 7th ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2013, pp. 9–16, http://dx.doi.org/10.1145/2507157.2507187.

[53] J. Yuan, F. Sivrikaya, F. Hopfgartner, A. Lommatzsch, M. Mu, Context-aware LDA: Balancing Relevance and Diversity in TV Content Recommenders, (2015). https://comcast.box.com/recsystv-2015-yuan (accessed April 30, 2018).

[54] C.-K. Hsieh, L. Yang, H. Wei, M. Naaman, D. Estrin, Immersive recommendation: News and event recommendations using personal digital traces, in: Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2016, pp. 51–62, http://dx.doi.org/10.1145/2872427.2883006.

[55] M. Allahyari, K. Kochut, Semantic context-aware recommendation via topic models leveraging linked open data, in: Web Information Systems Engineering – WISE 2016, Springer, Cham, 2016, pp. 263–277, http://dx.doi.org/10.1007/978-3-319-48740-3_19.

[56] L. Rabiner, B. Juang, An introduction to hidden Markov models, IEEE ASSP Mag. 3 (1986) 4–16, http://dx.doi.org/10.1109/MASSP.1986.1165342.

[57] S. Brooks, A. Gelman, G. Jones, X.-L. Meng, Handbook of Markov Chain Monte Carlo, CRC Press, 2011.

[58] M. Quadrana, P. Cremonesi, D. Jannach, Sequence-Aware Recommender Systems, ArXiv:1802.08452 [Cs]. (2018). http://arxiv.org/abs/1802.08452 (accessed June 11, 2018).

[59] N. Hariri, B. Mobasher, R. Burke, Context-aware music recommendation based on latenttopic sequential patterns, in: Proceedings of the Sixth ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2012, pp. 131–138, http://dx.doi.org/10.1145/2365952.2365979.

[60] M. Tavakol, U. Brefeld, Factored MDPs for detecting topics of user sessions, in: Proceedings of the 8th ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2014, pp. 33–40, http://dx.doi.org/10.1145/2645710.2645739.

[61] B. Huai, E. Chen, H. Zhu, H. Xiong, T. Bao, Q. Liu, J. Tian, Toward personalized context recognition for mobile users: A semisupervised Bayesian HMM approach, ACM Trans. Knowl. Discov. Data 9 (2014) 10:1–10:29, http://dx.doi.org/10.1145/2629504.

[62] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, X. Cheng, Learning Hierarchical Representation Model for NextBasket Recommendation, ACM Press, 2015, pp. 403–412, http://dx.doi.org/10.1145/2766462.2767694.

[63] M. Hosseinzadeh Aghdam, N. Hariri, B. Mobasher, R. Burke, Adapting recommendations to contextual changes using hierarchical hidden markov models, in: Proceedings of the 9th ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2015, pp. 241–244, http://dx.doi.org/10.1145/2792838.2799684.

[64] C. Trabelsi, S.B. Yahia, Harnessing the potential of HMM for movie rating recommendation, Procedia Comput. Sci. 96 (2016) 1543–1550, http://dx.doi.org/10.1016/j.procs.2016.08.201.

[65] D.A. Berry, B. Fristedt, Introduction, in: Bandit Problems, Springer, Dordrecht, 1985, pp. 1–8, http://dx.doi.org/10.1007/978-94-015-3711-7_1.

[66] S. Bubeck, N. Cesa-Bianchi, Regret analysis of stochastic and nonstochastic multi-armed bandit problems, Microsoft Res. 5 (1) (2017) https://www.microsoft.com/en-us/research/publication/regret-analysis-stochastic-nonstochastic-multi-armed-bandit-problems/ (accessed April 30, 2018).

[67] D. Bouneffouf, A. Bouzeghoub, A.L. Gançarski, A contextual-bandit algorithm for mobile context-aware recommender system, in: T. Huang, Z. Zeng, C. Li, C.S. Leung (Eds.), Neural Information Processing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 324–331, http://dx.doi.org/10.1007/978-3-642-34487-9_40.

[68] N. Hariri, B. Mobasher, R. Burke, Context adaptation in interactive recommender systems, in: Proceedings of the 8th ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2014, pp. 41–48, http://dx.doi.org/10.1145/2645710.2645753.

[69] L. Tang, Y. Jiang, L. Li, T. Li, Ensemble contextual bandits for personalized recommendation, in: Proceedings of the 8th ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2014, pp. 73–80, http://dx.doi.org/10.1145/2645710.2645732.

[70] L. Tang, Y. Jiang, L. Li, C. Zeng, T. Li, Personalized recommendation via parameter-free contextual bandits, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, 2015, pp. 323–332, http://dx.doi.org/10.1145/2766462.2767707.

[71] C. Zeng, Q. Wang, S. Mokhtari, T. Li, Online context-aware recommendation with time varying multi-armed bandit, in: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 2016, pp. 2025–2034, http://dx.doi.org/10.1145/2939672.2939878.

[72] Q. Wu, H. Wang, Q. Gu, H. Wang, Contextual bandits in a collaborative environment, in: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, 2016, pp. 529–538, http://dx.doi.org/10.1145/2911451.2911528.

[73] T.-Y. Liu, Learning to Rank for Information Retrieval, Springer-Verlag, Berlin Heidelberg, 2011, http://www.springer.com/gp/book/9783642142666 (accessed May 1, 2018).

[74] N.K. Tran, A. Ceroni, N. Kanhabua, C. Niederée, Back to the past: Supporting interpretations of forgotten stories by time-aware re-contextualization, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, ACM, New York, NY, USA, 2015, pp. 339–348, http://dx.doi.org/10.1145/2684822.2685315.

[75] A.Q. Macedo, L.B. Marinho, R.L.T. Santos, Context-aware event recommendation in event-based social networks, in: Proceedings of the 9th ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2015, pp. 123–130, http://dx.doi.org/10.1145/2792838.2800187.

[76] F. Yuan, G. Guo, J.M. Jose, L. Chen, H. Yu, W. Zhang, LambdaFM: Learning optimal ranking with factorization machines using lambda surrogates, ACM Press, 2016, pp. 227–236, http://dx.doi.org/10.1145/2983323.2983758.

[77] S. El Helou, C. Salzmann, D. Gillet, The 3A personalized, contextual and relation-based recommender system, J. UCS 16 (2010) 2179–2195.

[78] C. Ono, Y. Takishima, Y. Motomura, H. Asoh, Context-aware preference model based on a study of difference between real and supposed situation data, in: User Modeling, Adaptation, and Personalization, Springer, Berlin, Heidelberg, 2009, pp. 102–113, http://dx.doi.org/10.1007/978-3-642-02247-0_12.

[79] H.-S. Park, J.-O. Yoo, S.-B. Cho, A context-aware music recommendation system using fuzzy bayesian networks with utility theory, in: Fuzzy Systems and Knowledge Discovery, Springer, Berlin, Heidelberg, 2006, pp. 970–979, http://dx.doi.org/10.1007/11881599_121.

[80] J. Zhou, L. Ma, Q. Liu, L. Zhang, Y. Yu, Y. Pan, Minerva: A scalable OWL ontology storage and inference system, in: R. Mizoguchi, Z. Shi, F. Giunchiglia (Eds.), The Semantic Web – ASWC 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 429–443, http://dx.doi.org/10.1007/11836025_42.

[81] K. Oku, S. Nakajima, J. Miyazaki, S. Uemura, Context-aware SVM for context-dependent information recommendation, in: 7th International Conference on Mobile Data Management (MDM'06), 2006, http://dx.doi.org/10.1109/MDM.2006.56, pp. 109–109.
[82] M. Udell, C. Horn, R. Zadeh, S. Boyd, Generalized low rank models, Found. Trends Mach. Learn. 9 (2016) 1–118, http://dx.doi.org/10.1561/2200000055.
[83] M. Kagie, M. van Wezel, P.J.F. Groenen, Map based visualization of product catalogs, in: Recommender Systems Handbook, Springer, Boston, MA, 2011, pp. 547–576, http://dx.doi.org/10.1007/978-0-387-85820-3_17.
[84] C. Musto, G. Semeraro, P. Lops, M. de Gemmis, Random indexing and negative user preferences for enhancing content-based recommender systems, in: C. Huemer, T. Setzer (Eds.), E-Commerce and Web Technologies, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 270–281, http://dx.doi.org/10.1007/978-3-642-23014-1_23.
[85] L. Grasedyck, D. Kressner, C. Tobler, A literature survey of low-rank tensor approximation techniques, GAMM-Mitt. 36 (2013) 53–78, http://dx.doi.org/10.1002/gamm.201310004.
[86] M. Braunhofer, Hybridisation techniques for cold-starting context-aware recommender systems, in: Proceedings of the 8th ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2014, pp. 405–408, http://dx.doi.org/10.1145/2645710.2653360.
[87] V. Codina, F. Ricci, L. Ceccaroni, Semantically-enhanced pre-filtering for context-aware recommender systems, in: Proceedings of the 3rd Workshop on Context-Awareness in Retrieval and Recommendation, ACM, New York, NY, USA, 2013, pp. 15–18, http://dx.doi.org/10.1145/2442670.2442674.
[88] S. Inzunza, R. Juárez-Ramírez, S. Jiménez, User modeling framework for context-aware recommender systems, in: Recent Advances in Information Systems and Technologies, Springer, Cham, 2017, pp. 899–908, http://dx.doi.org/10.1007/978-3-319-56535-4_88.
[89] B.P. Knijnenburg, M.C. Willemsen, Z. Gantner, H. Soncu, C. Newell, Explaining the user experience of recommender systems, User Model. User-Adapt. Interact. 22 (2012) 441–504, http://dx.doi.org/10.1007/s11257-011-9118-4.
[90] Z. Yujie, W. Licai, Some challenges for context-aware recommender systems, in: 2010 5th International Conference on Computer Science Education, 2010, pp. 362–365, http://dx.doi.org/10.1109/ICCSE.2010.5593612.
[91] Y. Zheng, B. Mobasher, R. Burke, CARSKit: A java-based context-aware recommendation engine, in: 2015 IEEE International Conference on Data Mining Workshop (ICDMW), 2015, pp. 1668–1671, http://dx.doi.org/10.1109/ICDMW.2015.222.
[92] G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin, Incorporating contextual information in recommender systems using a multidimensional approach, ACM Trans. Inf. Syst. 23 (2005) 103–145, http://dx.doi.org/10.1145/1055709.1055714.
[93] A. Odić, M. Tkalčič, J.F. Tasič, A. Košir, Predicting and detecting the relevant contextual information in a movie-recommender system, Interact. Comput. 25 (2013) 74–90, http://dx.doi.org/10.1093/iwc/iws003.
[94] X. Ramirez-Garcia, M. García-Valdez, Post-filtering for a restaurant context-aware recommender system, in: Recent Advances on Hybrid Approaches for Designing Intelligent Systems, Springer, Cham, 2014, pp. 695–707, http://dx.doi.org/10.1007/978-3-319-05170-3_49.
[95] S. Consolvo, L. Arnstein, B.R. Franza, User study techniques in the design and evaluation of a ubicomp environment, in: G. Borriello, L.E. Holmquist (Eds.), UbiComp 2002: Ubiquitous Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 73–90, http://dx.doi.org/10.1007/3-540-45809-3_6.
[96] L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Lüke, R. Schwaiger, InCarMusic: Context-aware music recommendations in a car, in: C. Huemer, T. Setzer (Eds.), E-Commerce and Web Technologies, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 89–100, http://dx.doi.org/10.1007/978-3-642-23014-1_8.
[97] L. Baltrunas, B. Ludwig, S. Peer, F. Ricci, Context relevance assessment and exploitation in mobile recommender systems, Pers. Ubiquitous Comput. 16 (2012) 507–526, http://dx.doi.org/10.1007/s00779-011-0417-x.
[98] M. Elahi, M. Braunhofer, F. Ricci, M. Tkalcic, Personality-based active learning for collaborative filtering recommender systems, in: M. Baldoni, C. Baroglio, G. Boella, R. Micalizio (Eds.), AI*IA 2013: Advances in Artificial Intelligence, Springer International Publishing, Cham, 2013, pp. 360–371, http://dx.doi.org/10.1007/978-3-319-03524-6_31.
[99] L. Baltrunas, K. Church, A. Karatzoglou, N. Oliver, Frappe: Understanding the Usage and Perception of Mobile App Recommendations In-The-Wild, ArXiv:1505.03014 [Cs]. (2015). http://arxiv.org/abs/1505.03014 (accessed June 18, 2018).
[100] Y. Zheng, R. Burke, B. Mobasher, Differential context relaxation for context-aware travel recommendation, in: E-Commerce and Web Technologies, Springer, Berlin, Heidelberg, 2012, pp. 88–99, http://dx.doi.org/10.1007/978-3-642-32273-0_8.
[101] D. Parra, S. Sahebi, Recommender systems: Sources of knowledge and evaluation metrics, in: J.D. Velásquez, V. Palade, L.C. Jain (Eds.), Advanced Techniques in Web Intelligence-2, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 149–175, http://dx.doi.org/10.1007/978-3-642-33326-2_7.
[102] A. Batra, 21 Metrics for Measuring Online Display Advertising, (n.d.). http://webanalysis.blogspot.com/2014/05/21-metrics-for-measureing-online.html (accessed May 21, 2018).
[103] A. Gunawardana, G. Shani, Evaluating recommender systems, in: Recommender Systems Handbook, Springer, Boston, MA, 2015, pp. 265–308, http://dx.doi.org/10.1007/978-1-4899-7637-6_8.