Context Aware Recommender System Algorithms: State of the Art and Focus on Factorization Based Methods

Algorithmes de systèmes de recommandation sensibles au contexte: état de l'art et focalisation sur les méthodes basées sur la factorisation

Fatima Zahra Lahlou
ALBIRONI Research Team, ENSIAS, Mohamed V University, Rabat, Maroc
fatimazahra.lahlou@um5s.net.ma

Houda Benbrahim
ALBIRONI Research Team, ENSIAS, Mohamed V University, Rabat, Maroc
houda.benbrahim@um5.ac.ma

Ismail Kassou
ALBIRONI Research Team, ENSIAS, Mohamed V University, Rabat, Maroc
ismail.kassou@um5.ac.ma

Résumé

Les systèmes de recommandation sensibles au contexte (CARS) représentent un important champ de recherche depuis leur introduction en 2001 par (Herlocker and Konstan, 2001) et (Adomavicius and Tuzhilin, 2001). Selon Adomavicius et al. (Adomavicius and Tuzhilin, 2011), les algorithmes de CARS peuvent être classifiés selon trois principales catégories: pré-filtrage, post-filtrage et algorithmes de modélisation contextuelle. Étrangement, et jusqu'à l'année 2010, presque aucun algorithme de modélisation contextuelle n'a été proposé, même si les systèmes de recommandations basés sur la modélisation peuvent théoriquement accepter des variables supplémentaires (ici variables contextuelles) (Karatzoglou *et al.*, 2010). A partir de l'année 2010, plusieurs algorithmes de modélisation contextuelle de CARS ont été proposés, la plupart fondés sur la factorisation. Dans cet article, nous présentons d'abord, et suivant un ordre chronologique, l'état de l'art des algorithmes de CARS qui sont indépendants du domaine. Ensuite, nous étudions les modèles de factorisation utilisés et proposons quelques possibles directions de recherche afin de développer des algorithmes de modélisation contextuelle plus performants.

Abstract

Context Aware Recommender Systems (CARS) have become an important research area since its introduction in 2001 by (Herlocker and Konstan, 2001) and (Adomavicius and Tuzhilin, 2001). According to the classification of Adomavicius et al. (Adomavicius and Tuzhilin, 2011), there are three main categories of CARS algorithms: pre-filtering, post-filtering, and contextual modelling ones. Surprisingly, until the year of 2010, almost no CARS modelling algorithms were suggested, even though contextual modelling recommender systems can theoretically accept more dimensions as contextual variables (Karatzoglou et al., 2010). Starting from 2010, many contextual modelling CARS algorithms were suggested, most of them are built on factorization models. In this paper, we first present a state of the art of domain independent CARS algorithms listed following a chronological order. Then, we study factorization models used for the Context Aware Recommendation task and suggest some possible research directions for developing more performing contextual modelling CARS algorithms..

Mots-clés

Systèmes de Recommandation Sensibles au Contexte, Factorisation Matricielle, Factorisation de Tenseurs, Factorisation de Machines, Apprentissage Machine, Etat de l'Art.

Keywords

Context Aware Recommender Systems, Matrix Factorization, Tensor Factorization, Factorization Machines, Machine Learning, State of the Art.

1. Introduction

Recommender systems (RS) are systems that filter information depending on users' interests and suggest items to them that might match their preferences. Traditional Recommender Systems focus only on users and items when computing predictions. However, contextual information (such as time, weather, or accompanying persons) may influence user decisions. Indeed, the same item can be of interest to a user in a given context, and completely uninteresting in another one. For example, a user may book a hotel with business facilities for his/her business trip, and a different one with children's entertainment program for his/her family holiday. Therefore, contextual information should be considered in the recommendation process (Herlocker and Konstan, 2001) and (Adomavicius and Tuzhilin, 2001). Moreover, research proved that, in situation where context matters, including contextual information when computing recommendation improves its accuracy. Recommender Systems that consider contextual information are called Context Aware Recommender Systems (CARS).

The concept of "context" has been used in various disciplines as, Cognitive science, Linguistics, Philosophy, Psychology, Organizational Science, and Marketing in addition to Computer Science where it was studied in information retrieval, mobile computing, and e-commerce (Adomavicius and Tuzhilin, 2011). Therefore, many definitions of context have been suggested across these disciplines beyond dictionary definition that describes the context as: *"the influences and events related to a particular event or situation"* [1]. For the computing domain, and especially for CARS, the most cited definition is that of Dey *et al.* (Dey, 2001): *«Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.»*

The key idea of CARS is to consider, in addition to users and items, extra information related to context in order to provide more accurate predictions. To do that, a CARS's designer should answer to the main questions:

i) how to get contextual data?
ii) from what contextual data, which ones are relevant to recommendation purpose?
iii) and how to introduce contextual data in the recommendation process.

To obtain contextual data, Adomavicius *et al.* (Adomavicius and Tuzhilin, 2011) listed three different approaches:

- explicitly, by asking direct questions to relevant people or eliciting this data through other means;
- implicitly, from the data or the environment; for example, temporal information can be extracted from the time-stamp of a transaction, and information about user localization can be detected by a mobile phone company. This approach is common on in mobile context aware application as most current mobile devices are equipped with sensors that inform about users' current context;
- even by inference, using statistical or data mining methods. As examples, authors in (Lahlou *et al.*, 2013a, 2013b, 2013c) studied situations of "hotel booking" and "buying cars", and inferred from users' reviews contextual information "Trip type" and "First use of the car" respectively.

Adomavicius *et al.* (Adomavicius and Tuzhilin, 2011) described three paradigms to incorporate the contextual information into the recommendation algorithm:

- Contextual pre-filtering. Contextual information are used to filter relevant set of data, then recommendations are computed using any traditional recommender system on the selected data.
- Contextual post-filtering. Context is initially ignored, and recommendations are computed using the entire data, then, resulting recommendations are adjusted depending on contextual information.
- Contextual modelling, where contextual information are used directly in the modelling technique in order to provide predictions. Examples of algorithms belonging to these paradigms are mentioned in the section 2 of this paper.

Since the introduction of the CARS research issue in 2001 by (Herlocker and Konstan, 2001) and (Adomavicius and Tuzhilin, 2001), and until the year of 2010, best performing CARS algorithms were pre-filtering ones. However, almost no modelling algorithms were investigated, even though the model based recommender systems algorithms could theoretically accept more dimensions as contextual variables (Karatzoglou *et al.*, 2010). Later, and starting from 2010, number of contextual modelling CARS algorithms have been suggested and outperformed previous pre-filtering techniques. Most of these modelling algorithms are built on factorization models.

CARS were applied in many domains where contextual information can be captured and have a strong impact on users' decisions, such as touristic point of interest recommendation (Setten *et al.*, 2004), (Ardissono *et al.*, 2003),

[1] Cambridge Academic Content Dictionary, `http://dictionary.cambridge.org`

news recommendation (Garcin *et al.,* 2013); Zelenik and Bielikova, 2012), Search Personalisation (Kramár and Bieliková, 2012) and many others. Many research papers proposed algorithms for specific domains. Fewer tackled the issue of CARS algorithmic more globally and proposed new algorithms that are domain independent.
In this paper, we are interested in the second class of CARS algorithms that are domain independent. It is worth to note that, despite some papers presented a survey on context aware recommender systems, as it is the case for (Adomavicius and Tuzhilin, 2011) and (Liu *et al.* 2013), to the best of our knowledge, any of them tackled the specific case of domain independent CARS algorithms. Furthermore any of them studied in depth major factorization models used in this research issue as it is the case here.
In this paper, we first present a state of the art of domain independent CARS algorithms published since the beginning of CARS issue. We list them following a chronological order in the second section. This step allows us to notice that contextual modelling CARS algorithms started to truly develop until the year of 2010, and that most of them apply factorization models. This leads us to ask about the strength of factorization models for the Context Aware Recommendation task. Therefore, we present, in the third section, major factorization models used for the Context Aware Recommendation task. In the fourth section, we discuss these factorization models and suggest some possible research directions for developing more performing contextual modelling CARS algorithms. Finally, the last section concludes the paper.

2. History of CARS algorithms

Since the emergence of the idea of exploiting users' contexts in recommendation computation in 2001, a multitude of research papers were written, presenting each time new approaches and algorithms. Some of them are domain dependent, as it is the case for CARS papers on touristic point of interest recommendation (Setten *et al.,* 2004) (Ardissono *et al.,* 2003), news recommendation (Garcin *et al.,* 2013) (Zelenik and Bielikova, 2012), Search Personalisation (Kramár and Bieliková, 2012) and many others. Other papers tackle the issue of Context Aware Recommendation more globally and aim to develop new algorithms that can be used in any application domain. In this section, we are interested in this second category of research papers that we aim to trace the chronological evolution. To do so, we present major contributions in CARS algorithms following a chronological order.

2001: The concept of considering contextual data in rating prediction was introduced, almost at same time, by (Herlocker and Konstan, 2001) and (Adomavicius and Tuzhilin, 2001) in 2001, where the notion of context was called as a "task" in the first paper, and "multi-dimensions" in the other.
(Herlocker and Konstan, 2001) was almost the first to highlight that, computing recommendations solely based on user' historical ratings, assumes that user' interests are independent of the tasks at hand, which is not mostly the case. Researchers presented in this paper what they called a "task-focused approach". This approach consists in first asking the user to specify a task profile, then items related to this task profile are identified using correlation between items, and finally, resulting items are ranked dependently to interest prediction using a traditional recommender system based on historical ratings. One can consider this algorithm as a pre-filtering CARS one, following the previous classification of Adomavicius *et al.* (Adomavicius and Tuzhilin, 2011).
Almost at the same time, authors in (Adomavicius and Tuzhilin, 2001) attested again that it is not sufficient to recommend items to users, instead recommender systems should support additional dimensions, such as time or place. For this task, they suggested a multidimensional model to store additional contextual information together with user ratings, using the On-Line Analytical Processing (OLAP) data warehousing technique (Chaudhuri and Dayal, 1997). This work will be later extended in (Adomavicius *et al.,* 2005).
Although algorithms presented in (Herlocker and Konstan, 2001) and (Adomavicius and Tuzhilin, 2001) were the first research works that incorporated context into recommendation, one of their major contribution was to spotlight a new trend in RS research: Context Aware Recommendation.

2002-2004: Some CARS works were proposed (Ardissono *et al.,* 2003) (Setten *et al.,* 2004), but all of them were domain specific. At the best of our knowledge, no work on domain independent contextual modelling CARS algorithm were published within this period.

2005: Adomavicius *et al.* (Adomavicius *et al.,* 2005) followed the OLAP multidimensional data representation as presented in their previous work (Adomavicius and Tuzhilin, 2001), and use a rating estimation method called "reduction-based" approach. This method consists of computing predictions using only the ratings that pertain to the context of the user. This reduction-based approach had the advantage of allowing to use any traditional two-dimensional recommender. Note that this approach belongs to the pre-filtering class of CARS algorithms.
In another work, Chen (Chen, 2005) suggested a CARS approach based on user-user collaborative filtering algorithm, where the main idea is to compute first the similarity between contextual variables, and then incorporate

context into prediction by adapting the user-user collaborative filtering prediction formula, where the ratings in the formula are replaced by what the author called weighted rating. This last one is as a measure that computes a pseudo rating for an user u on an item i under a context c.
In other words, in the classical user-user collaborative filtering, rating prediction of the active user a on item i is expressed as:

$$p_{a,i} = \bar{r}_a + k \sum_{u=1}^{n} (r_{u,i} - \bar{r}_u) \cdot w_{a,u}$$

where n is the number of best neighbours chosen and k is a normalizing factor. Authors of (Chen, 2005) proposed to replace the ratings ($r_{u,i}$) in the expression above by some computed measure they called weighed rating ($R_{u,i,c}$):

$$p_{a,i,c} = \bar{r}_a + k \sum_{u=1}^{n} (R_{u,i,c} - \bar{r}_u) \cdot w_{a,u}.$$

Authors defined the weighted rating for a user a on item i with context c as the sum of ratings for the same user on the same item but on different contexts, weighted by the similarity between the context c and the other contexts.

2006: Oku *et al.* developed a first contextual modelling CARS algorithm: Context Aware Support Vector Machine (Oku *et al.,* 2006). The algorithm extends the Support Vector Machines (SVM) classifier, by adding axes of context to the feature space in order to consider the users' context. Authors conducted experiments showing that recommendations are improved when considering the context. However, using SVM for context aware recommendation is limited because of the high sparsity of data (Rendle *et al.,* 2011).

2009: Baltrunas and Ricci introduced a new CARS pre-filtering technique: the item splitting technique (Baltrunas and Ricci, 2009). This technique consists in, for each item whose ratings are significantly affected by a contextual variable, replacing this item by new items representing the couple (item, contextual variable). In other terms, one can consider that the item is split into sub-items, where each of them represent the item in a specific context.
Recommendations are then computed applying traditional recommendation algorithm on the resulting two-dimensional rating matrix, where original items are replaced by the new sub-items and contextual variables are omitted. Conducted experiments show that, when the splitting process results in homogeneous rating groups, (in other words when there are contextual variables that affect items ratings enough significantly) item splitting technique outperforms state of the art non-contextualized recommender systems. This work was extended later in (Baltrunas and Ricci, 2014). Also, other splitting approaches will be derived from the Item Splitting technique: User Splitting (Said *et al.,* 2011) and User Item Splitting (Zheng *et al.,* 2013).
Authors of (Panniello *et al.,* 2009) performed an experimental comparison between pre-filtering and post-filtering approaches across two datasets. For their comparison, they used as pre-filtering method what they call the *exact pre-filtering method*, which consists of using only the data that correspond to the specified context. They also conduct their study using two post-filtering methods: the *Weight* method, that reorders the recommender items depending on their relevance in the specific context; and the Filter method that filters out recommended items having small probability of relevance in the specific context. Their conducted experiments resulted in the fact that none of the methods always outperforms the other and so the best approach to use depends on the application itself.
It is interesting to observe that, until the year of 2010, even though the contextual modelling class of CARS algorithms could theoretically accept more dimensions as contextual variables, the only model that was investigated in this class was Support Vector Machines (SVM) (Oku *et al.,* 2006), as pointed out in (Karatzoglou *et al.,* 2010).

2010: Karatzoglou *et al.* developed a new CARS modelling algorithm based on Tensor Factorization and called Multi-verse Recommendation (Karatzoglou *et al.,* 2010). The intuition behind this algorithm is the same one behind Matrix Factorization used for traditional RS: factorize the rating matrix so as to model users, items and their interaction based on some latent features. Tensors are used, here, instead of matrices to represent multidimensional contextual rating matrix. Users, items, context, and their interactions are modelled using the High Order Singular Value Decomposition (HOSVD) method. Conducted experiments in (Karatzoglou *et al.,* 2010) showed that "Multi-verse Recommendation" outperformed not only standard non-contextual Matrix Factorization, but also up-to-date state-of the art context-aware recommendation approaches ((Adomavicius *et al.,* 2005) and (Baltrunas and Ricci, 2009)). Therefore, the strongest CARS algorithm in terms of prediction accuracy at that time was Multi-verse Recommendation. However, it has a high computational complexity: the number of model parameters to be learned grows exponentially with the number of contextual factors. Multi-verse Recommendation was the first factorization CARS algorithm to be developed.

2011: Baltrunas *et al.* suggested a Matrix Factorization approach for CARS instead of the precedent Tensor Factorization one, in order to overcome the large computation cost of Tensor Factorization method (Baltrunas

et al., 2011). To do so, the authors presented an algorithm that extends matrix factorization by introducing additional model parameters to model the interaction of contextual factors with item ratings. However, the Tensor Factorization algorithm (Karatzoglou *et al.,* 2010) remains the best one until that time.
In the meantime, Rendle *et al.* developed a new Recommender System algorithm: Factorization Machines (FM) (Rendle, 2010). All previous RS algorithms were designed first for a dense matrix, then applied on a sparse matrix (the rating matrix), where the sparsity was considered as a challenge to overcome. In contrast, the Factorization Machines algorithm starts from a sparse matrix, and use, to the best of our knowledge for the first time in RS, the sparse feature vectors representation, which allows considering the rating prediction problem as a common machine learning prediction task. Furthermore, FM overcomes the lack of data by using factorization. FM was successfully applied in various recommendation sub-tasks, including context aware recommendation in an algorithm called Context Aware Factorization Machines (CAFM) (Rendle *et al.,* 2011). CAFM outperform Multiverse Recommendation algorithm and becomes the best CARS predictor for that time.

2012: Tensor Factorization was again used for CARS in (Shi *et al.,* 2012), where the focus was the top-N recommendation for the case of implicit feedback scenarios. We talk about implicit feedback when users do not provide explicit ratings on evaluated items, but instead, the system collects traces of users' behaviours as click-streams, viewed items, time spent on an item page, etc. Unlike the case of explicit feedback, in implicit feedback scenarios the model parameters cannot be learned through minimizing the rating prediction error, simply because there are no explicit ratings. Authors in (Shi *et al.,* 2012) proposed a new CARS algorithm designed for implicit feedback scenarios, where they aim to maximize the Mean Average Precision (MAP) Top N-list evaluation metric. They hence utilize a Tensor Factorization approach to represent user-item-context interactions, and take the MAP evaluation technique into account for learning the model parameters. They called the proposed algorithm TFMAP.
In the same year, another work also studied the context aware implicit feedback case using a tensor factorization method (Hidasi and Tikk, 2012). Unlike explicit feedback case, implicit feedback tensor is large and dense, therefore, according to the authors, state of the art CARS methods, Multi-verse Recommendation (Karatzoglou *et al.,* 2010) and CAFM (Rendle *et al.,* 2011), cannot scale well. Authors proposed instead a new algorithm, called iTALS, which is a general ALS-based tensor factorization algorithm that scales linearly with the number of non-zeroes in the tensor and cubically with the number of features. It thus may be applied well on implicit data.

2014: Another contextual modelling CARS algorithm was proposed in (Nguyen *et al.,* 2014), called Gaussian Process Factorization Machines. Their authors argued that dominant state-of-the-art CARS approaches, Multi-verse Recommendation (Karatzoglou *et al.,* 2010) and Context Aware Factorization Machines (Rendle *et al.,* 2011), are limited because they model interactions between users, items, and contexts as some linear combination of their latent features. They attested that this may seem unrealistic to restrict these interactions to linearity, given the many possible types of interactions among them. They proposed a new CARS algorithm based on Gaussian Process where non-linear users-items-contexts interactions can be captured. The proposed algorithm used for the first-time Gaussian Process for CARS. It had also the particularity of being applicable to both explicit and implicit feedback.
In the meantime, Zheng *et al.* (Zheng *et al.,* 2014) (Zheng, 2014) assumed that it is difficult to interpret the latent features in matrix factorization based CARS algorithms. They aimed to develop instead a new CARS algorithm, that is good performing and easy to interpret the contextual effects. They propose to extend a non-contextual Recommender System approach, designed for the top-N subtask, called Sparse Linear Method (SLIM) (Ning and Karypis, 2011) which is based on the ItemKNN Collaborative Filtering. To do that, they learn from the data a sparse matrix of aggregation coefficients that are similar to the traditional item-item similarities. They thus develop a new CARS algorithm, called Contextual SLIM (CSLIM), where the idea is to incorporate the context into SLIM for top-N context aware recommendation. They investigated two different ways to incorporate context into SLIM: they first incorporated context by modelling contextual rating deviations (Zheng *et al.,* 2014), later, they integrated context similarity with SLIM following the intuition that recommendation list should be similar if contextual situations are similar (Zheng *et al.,* 2015).

2015: A different approach was used in (Liu *et al.,* 2015), where authors assumed that state-of-the-art CARS approaches, Multi-verse Recommendation, and Context Aware Factorization Machines, incorporate context into the factorization model by considering it as a dimension in the same way that they consider users and items, and do not model the semantic operation of context. Authors meant by the semantic operation of context some intuition inspired from Natural Language Programming research, which says that a context operate latent interests of users on items, in the same way that an adjective operates the latent vector of a noun representing its semantic information. They developed a new algorithm called Contextual Operating Tensor for CARS (COT) where they use a tensor to capture common effects of contexts, and generate a contextual operating matrix in order to compute rating predictions using an equation inspired from Matrix Factorization algorithm.

2016: Authors in (Codina *et al.*, 2016) developed a sophisticated pre-filtering algorithm based on context similarity called Distributional-Semantics Pre-filtering (DSPF). The algorithm adopts a pre-filtering approach, where the data that is most similar to the context of the active user is selected to compute recommendation using a traditional a two-dimension matrix factorization predictive model. Authors use a definition of similarity of contextual situations based on the distributional semantics of their composing conditions, following the intuition that situations are similar if they influence the user's ratings in a similar way.

In this section, we have presented following a chronological order, major research works on contextual modelling domain independent CARS algorithms. Note that the existing gaps between some years are due to the fact that we didn't find any work fitting these specifications for these specific years.

Works presented in this section are summarized following the classification of (Adomavicius and Tuzhilin, 2011) in table 1. By analysing table 1, one can observe that:

• until the year of 2010, only one contextual modelling CARS algorithm were proposed, whereas, starting from 2010, many contextual modelling CARS algorithms were developed;

• most of contextual modelling CARS algorithms are based on factorization models;

• the number of domain independent CARS works remain limited, thus further works can still be done in this research issue.

CARS techniques	CARS algorithms	Used Model (for contextual modelling techniques)
Contextual pre-filtering	(Herlocker and Konstan, 2001)	Not applicable
	Reduction based approach (Adomavicius and Tuzhilin, 2001) (Adomavicius *et al.*, 2005)	
	Item splitting technique (Baltrunas and Ricci, 2009) (Baltrunas and Ricci, 2014)	
	User splitting technique (Said *et al.*, 2011)	
	User Item splitting technique (Zheng *et al.*, 2013)	
	Distributional-Semantics Pre-filtering (Codina *et al.*, 2016)	
Contextual post-filtering	Weight and Filter post-filtering methods (Panniello *et al.*, 2009)	Not applicable
Contextual modelling	Context aware SVM (Oku *et al.*, 2006)	Support Vector Machines
	Multi-verse Recommendation (Karatzoglou *et al.*, 2010)	Tensor Factorization
	Context Aware Matrix Factorization (Baltrunas *et al.*, 2011)	Matrix Factorization
	Context Aware Factorization Machines (Rendle *et al.*, 2011)	Factorization Machines
	TFMAP (Shi *et al.*, 2012)	Tensor Factorization
	iTALS (Hidasi and Tikk, 2012)	Tensor Factorization
	Gaussian Process Factorization Machines (Nguyen *et al.*, 2014)	Gaussian Process
	Contextual SLIM (Zheng *et al.*, 2014) (Zheng, 2014) (Zheng *et al.*, 2015)	Sparse Linear Method
	Contextual Operating Tensor for CARS (Liu *et al.*, 2015)	Matrix Factorization

Table 1. Classification of domain independent CARS algorithms

In the remaining of the paper, we describe with more detail three major factorization models applied for CARS: Matrix Factorization, Tensor Factorization, and Context Aware Factorization Machines.

3. Factorization models for Context Aware Recommender Systems

3.1. Notations and Problem formulation

Classical RS aim to predict ratings for unobserved interactions between users and items. Thus, the goal is to define a target function on users $U = \{u_1, u_2, \dots u_m\}$ and items $I = \{i_1, i_2, \dots, i_n\}$:

$$y: U \times 1 \rightarrow \mathbb{R}$$

where *y(u,i)*is the rating of user *u* for item *i*. While only users and items are involved, this is called two-dimensional recommendation (Adomavicius *et al.,* 2005).

In contrast, CARS assume that additional information affect ratings like *Mood* or *Accompanying Person*. Thus, the rating function to estimate becomes:

$$y: U \times 1 \times C_1 \times C_2 \times \dots \times C_k \rightarrow \mathbb{R}$$

where C_1, C_2,…,C_k are the different sets of context that influence rating behavior.

For example, C_1 = {*happy, sad,* …} could be the Mood, C_2 = {*family, friends, business…*}could be the Accompanying Person and so on. Because additional dimensions are involved, CARS are called multi-dimensional recommendation (Adomavicius *et al.,* 2005).

Note that it is common in RS to represent ratings data in a matrix $R_{(m*n)}$, called rating matrix, where rows are users and columns are items and cells contains observed ratings. Because CARS involve additional information, CARS data are sometimes expressed as Tensors (Karatzoglou *et al.,* 2010). In the following, we introduce major factorization models designed for CARS. The figure bellow shows an example of how data can be modelled in a rating matrix.

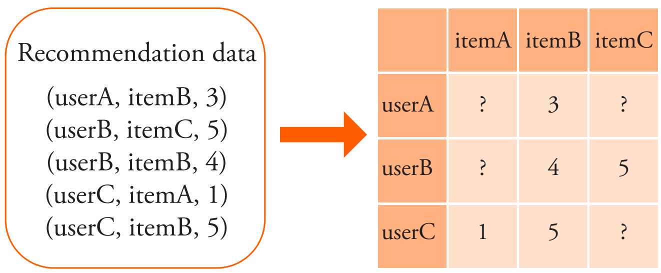

Figure 1. Rating matrix representation of recommendation data

3.2. Context Aware Matrix Factorization

3.2.1. Matrix Factorization for Non-Contextual Recommender Systems

One of the best performing algorithms for recommender systems is Matrix Factorization (MF) (Koren *et al.,* 2009). In its basic form, MF aims to factorize the rating matrix $R_{(m*n)}$ into two matrices $Q_{(m*r)}$ and $P_{(n*r)}$, so that:

$$R = Q^t * P$$

where Q represents P items and users, expressed on the basis of features vectors called latent factors. Representing users and items as vectors expressed in terms of these latent factor vectors is the key intuition behind matrix factorization algorithm: $Q_{item\_i,factor\_f}$ expresses how much the latent factor f is relevant for the item i, and $P_{user\_a,factor\_f}$ expresses how much the user a is interested by the latent factor f.

Predictions are then computed by multiplying item and user vectors:

$$\hat{r}_{u,i} = q_i^t p_u$$

Matrix factorization benefits from having a good scalability and predictive accuracy in addition to allowing the incorporation of additional information (Koren *et al.,* 2009). However, its major challenges lie on identifying the latent factors in addition to computing the mapping of each item and user to this latent factor vectors. To do so, it is very common to use the Singular Vector Decomposition (SVD) method belonging to Information Retrieval domain (Sarwar *et al.,* 2000). Applying SVD aim to decompose the rating matrix R into three matrices $U_{(m*r)}$, $S_{(r*r)}$ and $V_{(n*r)}$ so that:

$$R = U * S * V^t$$

where $U_{(m*r)}$ (Resp. $V_{(n*r)}$) is an orthogonal matrix representing Users (Resp. Items) in the basis of latent features and $S_{(r*r)}$ is a diagonal matrix representing the weight of each feature on the overall model.

Once matrices identified, the prediction computation is straightforward:

$$y(a,i) = \sum_{f=1}^{r} u_{a,f} * s_f * v_{i,f}$$

However, because the rating matrix is typically very sparse, applying SVD method needs adaptation. Some adaptations of SVD for the recommendation purpose were developed, among them, the Reduction Based SVD (Funk, 2006), also called Funk SVD, who suggest to use a regularized model, in order to avoid over-fitting and make then a better generalization. Following this method, rating prediction is computed as:

$$\hat{r}_{u,i} = \mu + b_u + b_i + q_i^t p_u$$

where μ is a general bias term, b_u and b_i represent bias for user and item, that is, User and item deviation from average, because all the ratings are not on the same scale. The loss function to minimize is the regularized squared error:

$$min_{p,q,b} \sum_{u,i} (r_{u,i} - q_i^t p_u - \mu - b_u - b_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2 + {b_u}^2 + {b_i}^2)$$

where λ is a regularization term and $r_{u,i}$ is the observed rating. Note that the sum is on the set of known ratings. Reduction Based SVD learn model parameters by minimizing the loss function, one feature at a time, using usually stochastic gradient descent.

Others adaptations of SVD for recommender systems were proposed like SVD++ (Koren, 2008) and Probabilistic Matrix Factorization (Salakhutdinov and Mnih, 2011), we detailed here only the Reduction Based SVD because its similarity with the factorization methods that we present above.

3.2.2. Matrix Factorization for CARS

Contextual dimensions were added to Matrix Factorization algorithm in (Baltrunas *et al.,* 2011) to build a new CARS algorithm called Context Aware Matrix Factorization (CAMF). The idea behind CAMF is to extend matrix factorization algorithm for RS by introducing additional model parameters to model the interaction between contextual conditions and ratings. Thus, the rating prediction is defined as:

$$\hat{r}_{u,i,c_1,\dots,c_k} = q_i^t p_u + b_u + b_i + \sum_{j=1}^{k} B_{ijc_j}$$

where $\hat{r}_{u,i,c1},\dots,c_k$ is the rating prediction for user *u* on item *i* under contexts $c_1,\dots,c_k$, and B_{ijcj} represent parameters modelling the interactions between contextual conditions and items. Researchers in (Baltrunas *et al.,* 2011) studied different level of interactions between contexts and items influencing the number of parameters B_{ijci}. Then, parameters are learned by minimizing a regularized squared error on training data. CAMF was compared to the best CARS algorithm at this time: Multi-verse Recommendation (MR) (Karatzoglou *et al.,* 2010), the reduction based approach (Adomavicius *et al.,* 2005) and the item splitting algorithm (Baltrunas and Ricci, 2009). Results show that, while CAMF and MR globally outperform other techniques, CAMF is the best algorithm when the context have a small influence on ratings, whereas MR is better than CAMF when the influence of context is stronger.

3.3. Context Aware Tensor Factorization

A tensor is a mathematical object that is derived from multi-linear algebra, a follow-up on linear algebra. It generalizes the vector and matrix concepts to multiple dimensions. The rank of the tensor is the number of its dimensions. Indeed, a tensor of rank zero is a scalar, a tensor of rank one is a vector and a tensor of rank two is a matrix. Tensors are important in domains where the need is to model data related to multiple dimensions.

In recommender systems, and as aforementioned, tensors are sometimes used to represent recommendation data, instead of rating matrices, especially in situations when recommendation data involves further dimensions than users and items. For instance, three dimensional tensors were used for tag-based recommender systems in (Rendle and Schmidt-Thieme, 2010) and time-aware recommender systems (Xiong *et al.,* 2010), representing user, item and tag (resp. user, item and time) dimensions. CARS are another field where it is convenient to use tensors, especially as multiple contextual information could be considered simultaneously (exp. mood, weather, accompanying person, …). In this case the usual two-dimensional rating matrix is converted into a multi-dimensional (user, item, context1, …, context k) tensor.

Tensor Factorization (TF) consist on factorizing the tensor into a lower dimensional vector space, so as, the original tensor is decomposed into lower rank tensors (and matrices). The first research work that was used tensors for CARS was Karatzoglou et al. in an algorithm called Multi-verse Recommendation (MR) (Karatzoglou *et al.,* 2010). The main intuition behind this algorithm was to mimic the matrix factorization method while taking additional dimensions into account. Indeed, authors try to model the variables by a reduced number of factors, while considering user-item-context interactions in the same way that users and items are modelled in Matrix Factorization techniques. To factorize the multidimensional rating tensor, authors use a technique called High Order Singular Value Decomposition (HOSVD). In this way, the contextual rating tensor is factorized into three matrices (representing respectively users, items and contexts) and one central tensor, as illustrated in the figure 2.

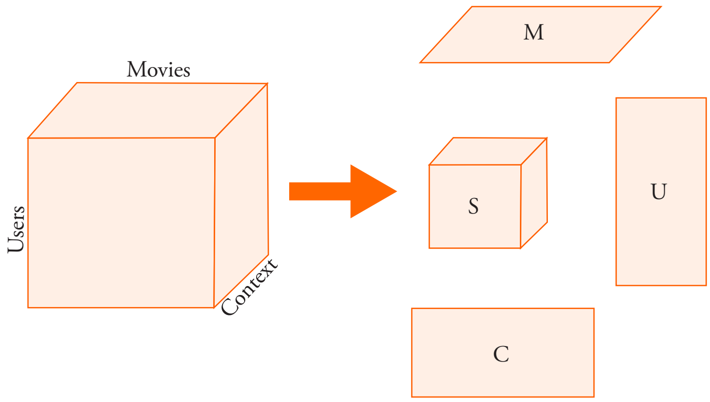

Figure 2. HOSVD factorization for a three-dimensional tensor (Karatzoglou et al., 2010).

Because HOSVD technique requires a dense matrix, authors adapted HOSVD by adding regularization and running the optimization procedure while considering only observed values in the rating tensor, as do the pre-mentioned Funk SVD algorithm. Conducted experiments on multiple datasets with different context influence show that MR consistently outperforms state of the art CARS techniques at that time: the reduction based approach (Adomavicius *et al.,* 2005) and the item splitting algorithm (Baltrunas and Ricci, 2009). MR become the best CARS algorithm for that time. However, MR has as limitation its high computational complexity: the number of model parameters to be learnt grows exponentially with the number of contextual factors.
Other works used tensor factorization for context aware recommendation (Hidasi and Tikk, 2012; Shi *et al.,* 2012). Authors in (Shi *et al.,* 2012) aim to maximize the Top N-list evaluation metric Mean Average Precision (MAP) in a CARS algorithm designed for implicit feedback scenarios. They use a Tensor Factorization approach to represent user-item-context interactions, and learn model parameters by considering the MAP evaluation technique. While researchers in (Hidasi and Tikk, 2012) tackle the issue of Tensor Factorization scalability, particularly for the case of implicit feedback were tensors are large and dense. They develop then a new CARS algorithm called iTALS, which is a general ALS-based tensor factorization algorithm that scales linearly with the number of non-zeroes in the tensor and cubically with the number of features.

3.4. Context Aware Factorization Machines

3.4.1. Factorization Machines

Factorization Machines algorithm (FM) (Rendle, 2010) starts from the Sparse Feature Vector data representation, then it applies the FM model where parameters are learned using an optimization procedure. In the following we explain Sparse Feature Vector data representation and FM model equation.

3.4.1.1. Sparse Feature Vector Data representation

As aforementioned, it is common in RS to represent data as a rating matrix, where columns are users and lines are items and cells contains observed ratings. Because not all users have rated all items, this matrix is typically very sparse. Note that tensors are also used to represent recommendation data when other dimensions are introduced, as is the case for context aware recommendation. In contrast to all previous works, Rendle (Rendle, 2010) use Sparse Feature Vector Representation to represent recommendation data instead of matrix.
Using Sparse Feature Vector Representation allows to represent data as a set of tuples (x,y) where $x \in \mathbb{R}^p$ is a real valued feature vector, $y \in \mathbb{R}$ the observed rating and $f(x) = y$. This representation has the particularity of expressing the recommendation problem as a common prediction problem formulation for machine learning, and then allow to apply standard machine learning methods. Sparse Feature vector representation has also the advantage of enabling to easily consider additional dimensions as contextual ones.
The figure 3 illustrate how data can be represented as Sparse Feature Vector instead of a rating matrix.

#	User	Movie	Rating
1	Alice	Titanic	5
2	Allice	Notting Hill	3
3	Alice	Star Wars	1
4	Bob	Star Wars	4
5	Bob	Star Trek	5
6	Charlie	Titanic	1
7	Charlie	Star Wars	5
…	…	…	…

	Feature vector x									Target y	
	User				Movie						
	A	B	C	…	TI	NH	SW	ST	…		
$X^{(1)}$	1	0	0	…	1	0	0	0	…	5	$y^{(1)}$
$X^{(2)}$	1	0	0	…	0	1	0	0	…	3	$y^{(2)}$
$X^{(3)}$	1	0	0	…	0	0	1	0	…	1	$y^{(3)}$
$X^{(4)}$	0	1	0	…	0	0	1	0	…	4	$y^{(4)}$
$X^{(5)}$	0	1	0	…	0	0	0	1	…	5	$y^{(5)}$
$X^{(6)}$	0	0	1	…	1	0	0	0	…	1	$y^{(6)}$
$X^{(7)}$	0	0	1	…	0	0	1	0	…	5	$y^{(7)}$

Figure 3. Example of recommendation data expressed using Sparse Feature Vectors Representation (Rendle, 2014)

3.4.1.2. Factorization Machines Model

As stated before, using Feature Vector Representation allows to apply machine learning methods for recommendation purpose. Although, they remain limited as far as data suffer from sparsity. For example, applying Linear Regression for rating prediction of a user *u* for an item *i* will end up to:

$$y(x) = w_0 + \sum_{i=1}^{p} w_i x_i$$

$$y(x) = w_0 + w_u + w_i$$

(because x_i is only non-zero for user *u* and item *i*). The resulting formula is easy to estimate but is not expressive enough because it outputs only the effect of the user and the item and not the user item interaction.
In contrast, applying Polynomial Regression for rating prediction of a user *u* for an item *i* will result in:

$$y(x) = w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p}\sum_{j\geq i}^{p} w_{i,j} x_i x_j$$

$$y(x) = w_0 + w_u + w_i + w_{u,i}$$

(because x_i is only non-zero for user *u* and item *i*). As one can see, this formula includes pairwise interactions but it is limited to observed ones. It cannot generalize for unobserved interactions because it cannot estimate the unobserved pairwise effects (Rendle, 2014).
Factorization Machines algorithm models interactions between the *p* input variables in *x* up to degree *d* using factorized interaction parameters. The degree *d* of FM represents the number of interactions between features to consider.
In this paper, we will limit on the second degree of FM since, in practice, FM is only used for pairwise interactions. Furthermore, in sparse settings, the case where FM are especially interesting to use, typically higher-order interactions are hard to estimate (Rendle, 2012).
The FM model equation of degree 2 is defined as:

$$y(x) = w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p}\sum_{j\geq i}^{p} < v_i, v_j > x_i x_j$$

where model parameters that have to be estimated are $w_0 \in \mathbb{R}$, $W \in \mathbb{R}^p$, $V \in \mathbb{R}^{p\times k}$, with w_0 the global bias, w_i the weight of i-th variable, and $<v_i, v_j>$ a dot product representing the interaction between the i-th and the j-th variable.
It is interesting to note that the model equation is similar to polynomial regression where pairwise interactions are factorized. This factorization has the main advantage of allowing to compute pairwise effect even for unobserved interactions.
Finally, an optimization procedure is used to estimates FM model parameters, such as Stochastic Gradient Descent (SGD) and others optimization procedures that are not in the scope of this paper (Rendle, 2012).

3.4.1.3. Factorization Machines Characteristics

In the following, we present some characteristics of FM that distinguish it from other RS algorithms:

- **FM algorithm is designed for a sparse matrix**. Previous RS algorithms are designed first for a dense matrix and then applied on the sparse rating matrix, where they consider the sparsity as a challenge to overcome. In opposite, Factorization Machines (FM) algorithm is designed at the beginning for a highly sparse matrix.
- **Factorize to estimate unobserved interactions**. As noticed before, in FM model, pair wise interactions are factorized. This factorization is a key element in FM algorithm, as it allows to compute pair wise effect even for unobserved interactions.
- **Generalization for other factorization approaches**. FM algorithm has also the particularity of generalizing other factorization approaches. As shown in (Rendle, 2010), FM can mimic several factorization models just by an appropriate definition of the input vector x using binary indicator variables. For example, applying the FM model for a basic two-dimensional recommendation task using Sparse Feature Vector representation for rating prediction of a user for an item will lead to:

$$y(x) = w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p} \sum_{j \geq i}^{p} < v_i, v_j > x_i x_j$$

$$y(x) = w_0 + w_u + w_i + <w_{ui} v_i>$$

$$y(x) = w_0 + w_u + w_i + <w_u^T v_i>$$

(because x_i is only non-zero for user u and item I).

One can observe that the last formula is identical to Matrix Factorization one. More example of generalization of RS factorization approaches are detailed in (Rendle, 2010). We omitted these approaches because they are related to other RS tasks that are not in the scope of this paper.

3.4.2. Factorization Machines for CARS

FM algorithm was applied to the Context Aware task in a new CARS algorithm called Context Aware Factorization Machines (CAFM) (Rendle *et al.,* 2011).

CAFM did not need any tuning from its origin FM. Indeed, using Sparse Vector Representation enable to easily consider additional dimensions, as context, without any transformation. Furthermore, the use of Sparse Vector Representation allows to model different types of contexts, as categorical context (e.g. mood), or set categorical (e.g. last watched movies) or even real-valued contexts (e.g. time) just by adapted encoding of variable in the data representation.

Note that in Tensor Factorization (Karatzoglou *et al.,* 2010) only categorical context can be modelled. Note also that FM algorithm factorizes all pairwise interactions with all contextual variables (Rendle *et al.,* 2011).

The figure 4, taken from (Rendle *et al.,* 2011), shows an example of contextual recommendation data expressed using Sparse Feature Vector Representation.

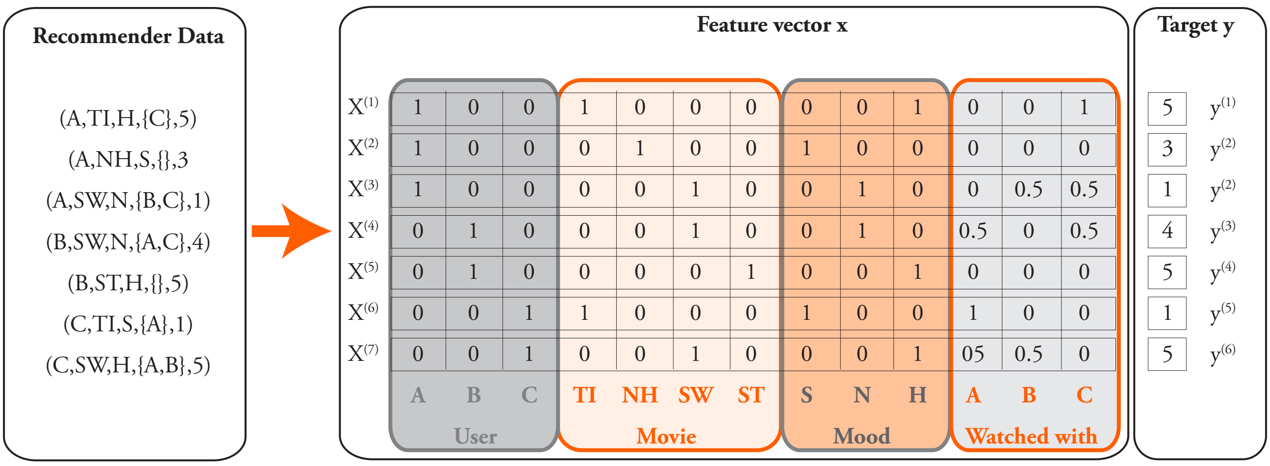

Figure 4. Example of contextual recommendation data expressed using Sparse Feature Vector Representation

CAFM were compared to the best performing CARS algorithm in that time, that was MR (Karatzoglou *et al.,* 2010). Experiments show that on dense datasets, the two algorithms have comparable prediction quality. But on sparse datasets, FM outperforms MR. Furthermore, FM is better in terms of runtime (Rendle *et al.,* 2011).

4. Discussion and Open Future Research Directions

As aforementioned, the algorithms detailed in the previous section were criticized by some succeeding works:

i) authors of (Nguyen *et al.,* 2014) found unrealistic that MR and CAMF restrict interactions between users, items and contexts to linearity, given the many possible types of interactions among them;
ii) whereas authors of (Liu *et al.,* 2015) assume that MR and CAMF incorporate context into the factorization model by considering it as a dimension in the same way that they consider users and items;
iii) while in (Shi *et al.,* 2012), researchers stressed that previous works are limited to the explicit feedback scenarios and do not consider the case where feedback are implicit. Nevertheless, these algorithms have a real strength, not only because of their good performance, but mainly because they started the research in the modelling CARS algorithms. In this section, we discuss what may be reason of their strength.

Context Aware Matrix Factorization (CAMF) gives good results compared to non-factorization CARS techniques. As cited before, CAMF extend matrix factorization algorithm for RS by introducing additional model parameters to model the interaction between contextual factors and ratings. One can observe that this algorithm factorizes only users and items, but not context. This is maybe the reason why CAMF is outperformed by Tensor Factorization, as this last algorithm factorize also context.

Multi-verse Recommendation (MR) algorithm gives better results for the context aware recommendation task. Recall that the intuition behind MR is the same one behind matrix factorization algorithm, one of the best Recommender Systems algorithms: factorizing the rating Tensor in order to model users, items, and contexts by considering interactions between them. One can observe that, unlike CAMF, contexts are also factorized, it should be the reason why MR outperforms CAMF.

However, the best CARS algorithm among the three remains Context Aware Factorization Machines (CAFM), especially on sparse dataset where it outperforms MR. Recall that, Like MR algorithm, Context Aware Factorization Machines factorize also the contexts. However, unlike MR and all other RS algorithms, FM algorithm was designed for sparse data. The second main advantage of Context Aware Factorization Machines is its principle to factorize to estimate unobserved interactions, which play a consequent role to overcome the sparsity challenge. It is important to highlight that the sparsity in recommendation data is the reason behind the failure of classical machine learning algorithm, like linear or polynomial regression. Furthermore, the concept of latent factor behind factorization methods, as Matrix Factorization and Tensor Factorization, is the key of success of these methods. FM combined between both the concept of latent factors and polynomial regression in such a way to overcome the sparsity limitation. It is the third reason behind its strength.

Research in contextual modelling context aware recommender has made important progress, especially with the emergence of factorization algorithms. However, there are still some research directions to build more powerful models:

- First, the seemingly best CARS algorithm, CAFM, is in reality a general recommender system algorithm that can be applied on context aware task without need of any special adaptation. A possible research direction could be to rethink CAFM in such a way to consider special characteristics that are proper to the context.
- Furthermore, and as mentioned in (Liu *et al.,* 2015), contextual dimension is treated in the same way as user and item. Another research direction could be to develop new model where contextual information is used in such a way to adapt ratings when context matter, while users and items should be the core of the model.
- Another possible direction to investigate is using one of the well performing pre-filtering methods, like the item splitting technique (Baltrunas and Ricci, 2009). One can consider to combine these pre-filtering methods with some modelling algorithms in order to benefit from its strength and develop a more powerful model.
- One can also consider other forms of users-items-context interactions than linearity as done by (Nguyen *et al.,* 2014).

5. Conclusion

Context Aware Recommender Systems received researchers' interests since its beginning in 2001. Although considering additional dimensions theoretically is not a big challenge to model based recommender systems algorithms, contextual modelling CARS algorithms had taken time to really start to be developed. In this paper, we have presented the historical evolution of CARS algorithms. We have also deeply detailed some major factorization CARS algorithms and discuss their strengths and limitations. The emergence of factorization CARS algorithms has made a big progress in CARS algorithmic. Factorization Machines, particularly, has assets that enable it to overcome the sparsity challenge of contextual data. However, progress still needs to be made in the research area, and new directions should be explored to develop more powerful contextual modelling CARS algorithms.

6. References

Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A., 2005. *Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach*. ACM Trans Inf Syst 23, 103–145. doi:10.1145/1055709.1055714

Adomavicius, G., Tuzhilin, A., 2011. *Context-aware recommender systems*. In Recommender Systems Handbook. Springer, pp. 217–253.

Adomavicius, G., Tuzhilin, A., 2001. *Multidimensional Recommender Systems: A Data Warehousing Approach*. In Fiege, L., Mühl, G., Wilhelm, U. (Eds.), Electronic Commerce, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 180–192. doi:10.1007/3-540-45598-1_17

Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P., 2003. Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices. Appl. Artif. Intell. 17, 687–714. doi:10.1080/713827254

Baltrunas, L., Ludwig, B., Ricci, F., 2011. *Matrix Factorization Techniques for Context Aware Recommendation.* In Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11. ACM, New York, NY, USA, pp. 301–304. doi:10.1145/2043932.2043988

Baltrunas, L., Ricci, F., 2009. *Context-based Splitting of Item Ratings in Collaborative Filtering*. In Proceedings of the Third ACM Conference on Recommender Systems, RecSys '09. ACM, New York, NY, USA, pp. 245–248. doi:10.1145/1639714.1639759

Baltrunas, L., Ricci, F., 2014. *Experimental evaluation of context-dependent collaborative filtering using item splitting*. User Model. User-Adapt. Interact. 24, 7–34. doi:10.1007/s11257-012-9137-9

Chaudhuri, S., Dayal, U., 1997. *An Overview of Data Warehousing and OLAP Technology*. SIGMOD Rec 26, 65–74. doi:10.1145/248603.248616

Chen, A., 2005. *Context-Aware Collaborative Filtering System: Predicting the User's Preference in the Ubiquitous Computing Environment*. In Strang, T., Linnhoff-Popien, C. (Eds.), Location- and Context-Awareness, Lecture Notes in Computer Science. Presented at the International Symposium on Location- and Context-Awareness, Springer Berlin Heidelberg, pp. 244–253. doi:10.1007/11426646_23

Codina, V., Ricci, F., Ceccaroni, L., 2016. *Distributional semantic pre-filtering in context-aware recommender systems*. User Model. User-Adapt. Interact. 26, 1–32. doi:10.1007/s11257-015-9158-2

Dey, A.K., 2001. *Understanding and using context*. Pers. Ubiquitous Comput. 5, 4–7.

Funk, S., 2006. Netflix update: Try this at home.

Garcin, F., Dimitrakakis, C., Faltings, B., 2013. *Personalized News Recommendation with Context Trees*. In Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13. ACM, New York, NY, USA, pp. 105–112. doi:10.1145/2507157.2507166

Herlocker, J.L., Konstan, J.A., 2001. *Content-Independent Task-Focused Recommendation*. IEEE Internet Comput. 5, 40–47. doi:10.1109/4236.968830

Hidasi, B., Tikk, D., 2012. *Fast ALS-Based Tensor Factorization for Context-Aware Recommendation from Implicit Feedback*. In Flach, P.A., Bie, T.D., Cristianini, N. (Eds.), Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science. Presented at the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, pp. 67–82. doi:10.1007/978-3-642-33486-3_5

Karatzoglou, A., Amatriain, X., Baltrunas, L., Oliver, N., 2010. *Multiverse Recommendation: N-dimensional Tensor Factorization for Context-aware Collaborative Filtering*. In Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10. ACM, New York, NY, USA, pp. 79–86. doi:10.1145/1864708.1864727

Kramár, T., Bieliková, M., 2012. *Dynamically Selecting an Appropriate Context Type for Personalisation*. In Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12. ACM, New York, NY, USA, pp. 321–324. doi:10.1145/2365952.2366032

Koren, Y., 2008. *Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model*. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08. ACM, New York, NY, USA, pp. 426–434. doi:10.1145/1401890.1401944

Koren, Y., Bell, R., Volinsky, C., 2009. Matrix factorization techniques for recommender systems. Computer 42, 30–37.

Lahlou, F.Z., Benbrahim, H., Mountassir, A., Kassou, I., 2013a. *Inferring Context from Users' Reviews for Context Aware Recommendation*. In: Bramer, M., Petridis, M. (Eds.), Research and Development in Intelligent Systems XXX. Springer International Publishing, pp. 227–239.

Lahlou, F.Z., Benbrahimand, H., Mountassir, A., Kassou, I., 2013b. *Context extraction from reviews for Context Aware Recommendation using Text Classification techniques*. In: 2013 ACS International Conference on Computer Systems and Applications (AICCSA). Presented at the 2013 ACS International Conference on Computer Systems and Applications (AICCSA), pp. 1–4. doi:10.1109/AICCSA.2013.6616512

Lahlou, F.Z., Mountassir, A., Benbrahim, H., Kassou, I., 2013c. *A Text Classification based method for context extraction from online review.*, In: 2013 8th International Conference on Intelligent Systems: Theories and Applications (SITA). Presented at the 2013 8th International Conference on Intelligent Systems: Theories and Applications (SITA), pp. 1–5. doi:10.1109/SITA.2013.6560804

Liu, Q., Ma, H., Chen, E., & Xiong, H., 2013. *A survey of context-aware mobile recommendations*. International Journal of Information Technology & Decision Making, 12(01), 139-172.

Liu, Q., Wu, S., Wang, L., 2015. *COT: Contextual Operating Tensor for Context-Aware Recommender Systems*. In: AAAI. pp. 203–209.

Nguyen, T.V., Karatzoglou, A., Baltrunas, L., 2014. *Gaussian Process Factorization Machines for Context-aware Recommendations*. In Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '14. ACM, New York, NY, USA, pp. 63–72. doi:10.1145/2600428.2609623

Ning, X., Karypis, G., 2011. *SLIM: Sparse Linear Methods for Top-N Recommender Systems*. In 2011 IEEE 11th International Conference on Data Mining. Presented at the 2011 IEEE 11th International Conference on Data Mining, pp. 497–506. doi:10.1109/ICDM.2011.134

Oku, K., Nakajima, S., Miyazaki, J., Uemura, S., 2006. *Context-Aware SVM for Context-Dependent Information Recommendation.* In Proceedings of the 7th International Conference on Mobile Data Management, MDM '06. IEEE Computer Society, Washington, DC, USA, p. 109–. doi:10.1109/MDM.2006.56

Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., Pedone, A., 2009. *Experimental Comparison of Pre- vs. Post-filtering Approaches in Context-aware Recommender Systems*. In Proceedings of the Third ACM Conference on Recommender Systems, RecSys '09. ACM, New York, NY, USA, pp. 265–268. doi:10.1145/1639714.1639764

Paterek, A., 2007. *Improving regularized singular value decomposition for collaborative filtering*. In Proceedings of KDD Cup and Workshop. pp. 5–8.

Rendle, S., 2014. *Factorization Machines. In Data Mining (ICDM)*. 2010 IEEE 10th International Conference on (pp. 995-1000). IEEE.

Rendle, S., 2012. *Factorization Machines with libFM*. ACM Trans Intell Syst Technol 3, 57:1–57:22. doi:10.1145/2168752.2168771

Rendle, S., 2010. *Factorization Machines*. In 2010 IEEE International Conference on Data Mining. Presented at the 2010 IEEE International Conference on Data Mining, pp. 995–1000. doi:10.1109/ICDM.2010.127

Rendle, S., Gantner, Z., Freudenthaler, C., Schmidt-Thieme, L., 2011. *Fast Context-aware Recommendations with Factorization Machines*. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11. ACM, New York, NY, USA, pp. 635–644. doi:10.1145/2009916.2010002

Rendle, S., Schmidt-Thieme, L., 2010. *Pairwise Interaction Tensor Factorization for Personalized Tag Recommendation*, In Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM '10. ACM, New York, NY, USA, pp. 81–90. doi:10.1145/1718487.1718498

Said, A., De Luca, E.W., Albayrak, S., 2011. *Inferring contextual user profiles-improving recommender performance*. In Proceedings of the 3rd RecSys Workshop on Context-Aware Recommender Systems.

Salakhutdinov, R., Mnih, A., 2011. *Probabilistic matrix factorization*. In NIPS. pp. 1–8.

Sarwar, B., Karypis, G., Konstan, J., Riedl, J., 2000. *Application of Dimensionality Reduction in Recommender System - A Case Study (No. TR-00-043)*. Minnesota Univ Minneapolis Dept of Computer Science.

Setten, M. van, Pokraev, S., Koolwaaij, J., 2004. *Context-Aware Recommendations in the Mobile Tourist Application COMPASS*. In Bra, P.M.E.D., Nejdl, W. (Eds.), Adaptive Hypermedia and Adaptive Web-Based Systems, Lecture Notes in Computer Science. Presented at the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Springer Berlin Heidelberg, pp. 235–244. doi:10.1007/978-3-540-27780-4_27

Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., Hanjalic, A., Oliver, N., 2012. *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, SIGIR '12. ACM, New York, NY, USA, pp. 155–164. doi:10.1145/2348283.2348308

Xiong, L., Chen, X., Huang, T.-K., Schneider, J., Carbonell, J.G., 2010. *Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization*. In Parthasarathy, S., Liu, B., Goethals, B., Pei, J., Kamath, C. (Eds.), Proceedings of the 2010 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, Philadelphia, PA, pp. 211–222.

Zelenik, D., Bielikova, M., 2012. *Reducing the Sparsity of Contextual Information for Recommender Systems*. In Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12. ACM, New York, NY, USA, pp. 341–344. doi:10.1145/2365952.2366037

Zheng, Y., 2014. *Deviation-based and Similarity-based Contextual SLIM Recommendation Algorithms*. In Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14. ACM, New York, NY, USA, pp. 437–440. doi:10.1145/2645710.2653368

Zheng, Y., Mobasher, B., Burke, R., 2015. *Integrating Context Similarity with Sparse Linear Recommendation Model*. In Ricci, F., Bontcheva, K., Conlan, O., Lawless, S. (Eds.), User Modelling, Adaptation and Personalization, Lecture Notes in Computer Science. Presented at the International Conference on User Modelling, Adaptation, and Personalization, Springer International Publishing, pp. 370–376. doi:10.1007/978-3-319-20267-9_33

Zheng, Y., Mobasher, B., Burke, R., 2014. *Deviation-Based Contextual SLIM Recommenders*. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14. ACM, New York, NY, USA, pp. 271–280. doi:10.1145/2661829.2661987

Zheng, Y., Mobasher, B., Burke, R.D., 2013. *The Role of Emotions in Context-aware*.