Received December 17, 2018, accepted January 16, 2019, date of publication February 4, 2019, date of current version March 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2897003

Kernel Context Recommender System (KCR): A Scalable Context-Aware Recommender System Algorithm

MISBAH IQBAL[1], MUSTANSAR ALI GHAZANFAR[2], ASMA SATTAR[3], MUAZZAM MAQSOOD[4], SALABAT KHAN[4], IRFAN MEHMOOD[5], AND SUNG WOOK BAIK[6], (Member, IEEE)

[1]BPP Business School, BPP University, London 143-747, U.K.
[2]School of Architecture, Computing and Engineering, University of East London, London E16 2RD, U.K.
[3]Department of Computer Science, FAST-National University of Computer and Emerging Sciences, Faisalabad 44000, Pakistan
[4]Department of Computer Science, COMSATS University Islamabad, Attock Campus, Attock 43600, Pakistan
[5]Department of Software, Sejong University, Seoul 05006, South Korea
[6]Intelligent Media Laboratory, Digital Contents Research Institute, Sejong University, Seoul 05006, South Korea

Corresponding author: Irfan Mehmood (irfanmehmood@ieee.org)

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIP), under Grant No. 2018R1C1B5086294.

ABSTRACT Recommender systems are intelligent data mining applications that deal with the issue of information overload significantly. The available literature discusses several methodologies to generate recommendations and proposes different techniques in accordance with users' needs. The majority of the work in the recommender system domain focuses on increasing the recommendation accuracy by employing several proposed approaches where the main motive remains to maximize the accuracy of recommendations while ignoring other design objectives, such as a user's an item's context. The biggest challenge for a recommender system is to produce meaningful recommendations by using contextual user-item rating information. A context is a vast term that may consider various aspects; for example, a user's social circle, time, mood, location, weather, company, day type, an item's genre, location, and language. Typically, the rating behavior of users varies under different contexts. From this line of research, we have proposed a new algorithm, namely Kernel Context Recommender System, which is a flexible, fast, and accurate kernel mapping framework that recognizes the importance of context and incorporates the contextual information using kernel trick while making predictions. We have benchmarked our proposed algorithm with pre- and post-filtering approaches as they have been the favorite approaches in the literature to solve the context-aware recommendation problem. Our experiments reveal that considering the contextual information can increase the performance of a system and provide better, relevant, and meaningful results on various evaluation metrics.

INDEX TERMS Context, context-aware kernel mapping recommender systems, recommender system kernel.

I. INTRODUCTION

A. RECOMMENDER SYSTEMS

In this digital era, the Internet has become the prominent essential of life. People are using various platforms on the Internet to entertain their different needs and activities such as shopping, watching videos, education, communication, following their favorite celebrities, generating new trends, business, entrepreneurship and many more activities. These all activities are not only producing huge data but also demanding a structured approach to access the information that is fast, reliable and relevant. This tremendous amount of data, for example, music (LastFM), Spotify (spotify.com), Pandora (pandora.com), movies and videos (e.g. in Netflix, netix.com), in YouTube (youtube.com), online services (e.g. Amazon, amazon.com), Flicker (flickr.com) is causing an information overload in the digital domain. Due to this phenomenon of information overload, the need for effective filtration techniques to access the relevant data has become unavoidable.

The associate editor coordinating the review of this manuscript and approving it for publication was Sudhakar Babu.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Recommender systems are intellectual applications that mitigate the information overload issues to a great extent by filtering according to the user's requirement [1]. The Group Lens System [2] is an example of recommender systems. It proposes that a user must read a Usenet News article and give their opinion. Then, these opinions are used as ratings and people with similar rating behavior (like-minded people or similar users), help the system to predict ratings for each other. Varieties of such systems are available nowadays, but they are not flexible enough because the quality of recommendation algorithms depends on different factors. The fundamental point of these intelligent frameworks is to give significant and expressive recommendations about items, to the different users according to their requirements and interests. Different search engines are available that performs information filtering. Search engines filter out different pages or information through the explicit queries given by internet users, but the comprehensive objectives (of suggesting users what they want) are not fulfilled by these engines. It is very difficult to pinpoint the user's current needs by using simple keywords. The Semantic Web is another alternative that provides information filtering based on how web pages are interpreted or annotated but it is very difficult to annotate web pages. All such problems have emphasized the need for a system that not only filters out the information for the user but also predicts whether the user would like a given resource.

TABLE 1. Symbol representations.

Symbols	Representation
$\mathcal{U}$	List of users
$\mathcal{I}$	List of items
M	Number of users in the system
N	Number of items in the system
r_{iu}	Rating given by user u on item i
$\mathcal{D}$	The set of user-item pairs that have been rated

A recommender system is based on two entities: 'users' and 'items'. These users provide ratings about the items and then the recommender system provides recommendations about the new items based on the ratings provided by the Users. In this work, users are denoted by $\mathcal{U} = \{u_1, u_2, \cdots, u_M\}$, where total number of users using the system is $|\mathcal{U}| = M$, and denote the set of items being recommended by $\mathcal{I} = \{ i_1, i_2, \cdots, i_N\}$, with $|\mathcal{I}| = N$. The users will have rated some, but not all, of the items. Ratings are denoted by $(r_{iu}|(i, u) \in \mathcal{D})$, where $\mathcal{D} \subset \mathcal{I} \times \mathcal{U}$ is the set of user-item pairs that have been rated. A total number of ratings made are denoted by$|\mathcal{D}| = T$. Typically each user rates only a small number of the possible items, so that $|\mathcal{D}| = T \ll |\mathcal{I} \times \mathcal{U}| = N \times M$. It is not unusual in practical systems to have $T/(N \times M) \approxeq 0.01$. The set of possible ratings made by the users can be thought of as elements of an $M \times N$ rating matrix R. We denote the items for which there are ratings by user u as $\mathcal{D}_u$, and the users who have rated an item i by $\mathcal{D}_i$. The task is to create a recommendation algorithm that predicts an unseen rating r_{iu}, i.e. for $(i, u) \notin \mathcal{D}$. Main description of basic symbols used in this sections is given in Table 1.

There has been a lot of research approaches proposed for recommender systems. These approaches include collaborating filtering [3], content-based filtering [4], Knowledge (Ontology) based filtering [5], [6], and demographic-based filtering [7]. Moreover, hybrid recommender system approaches have also been proposed by combining individual approaches to handle the shortcomings of the aforementioned approaches [8]–[10].

The literature of recommender system ranges from memory-based approaches [3], [11] to model-based approaches [12]–[14] spanning a number of algorithms, where the main motif remains to increase the accuracy of the recommender system over a user-item rating matrix while ignoring other important design objectives, such as the context in which a user is rating an item. In recent studies, the concept of cross-domain CF has been introduced which is used to solve the sparsity problem in Collaborative Filtering. Yu *et al.* [15] propose a user based Cross Domain CF algorithm, which is based on the Linear Decomposition Model (CDCFLDM). They used a linear decomposition technique to find out the total and local similarity. Furthermore, in Yu *et al.* [16] also worked on cross-domains and created a two-sided Cross-Domain CF Model, which covers two auxiliary domains that is user-sided domain and item-sided domain. In order to perform this, they first implemented the bi-orthogonal tri-factorization model to extract the intrinsic features of both users and items for both domains. Then these domain independent features are used to generate feature vectors. User-item ratings and interactions were then used for trainings purpose.

In this work, considering the importance of the contextual information, we propose a novel context-aware kernel-based recommendation algorithm that builds model over user-item context rating matrix.

B. CONTEXT-AWARE RECOMMENDER SYSTEMS

Different researchers have described the term context in different ways. A context is a generic term; it may include location, environment, social networks, age, gender, history, mood, time (morning, evening, noon) etc. In order to improve the users experience and for the comfort of the users, the recommender systems should adapt to the user's context. The context can be described as a term or information that is related to the user's interaction with the system; for example, time, history, location, on individuality, social context, activities, the genre of the movie/music. Certain contexts such as, the mood (happy, sad) of a user when they rated an item, location, time (morning, evening, noon) when an item is rated, history of items that a user rated (or showed interest in) have a great influence over rating behavior. Rendle *et al.* [17] formulates the main difference of the user-item characteristics and context is that these attributes are related to just an item that how this item has been rated but context is linked with the whole scenario while rating (e.g. the mood of the user when rating an item, the time and the interests of a user at that particular time).

In order to incorporate contextual information into the basic recommender systems, a new sub-field of recommender systems has been emerged known as *Context-Aware Recommender systems* [18], [19]. Context-aware recommender systems are the systems that consider all such contextual information while processing recommendations. These systems provide recommendations that are more relevant to the user's needs and preferences. Contextual information also has a great influence over the rating behavior of the user.

In this paper, we mainly focus on context-aware recommender systems and investigate the effects of contextual information on the performance of the recommender system.

The major contributions of this paper are highlighted as following:

1) We propose a Context-Aware Kernel Mapping Recommender (KCR) system Algorithm, which uses various user-related (10) and item related (5) contexts for both item-based and user-based versions of the algorithm. Empirically, we show that the proposed context aware algorithm (KCR) has better performance than other existing recommender systems, which ultimately explains the need of context aware RS.
2) We propose Additive and Multiplicative Models for both user- and item-based versions. This paper also describes how more information (Context Kernels) can be added linearly and non-linearly and its effect on recommendations quality.
3) We show how variation in Kernel types such as Ploy-Gaussian, Polynomial etc. has an impact on context-aware predictions.
4) We also generated the post-filtered model of our KCR Algorithm in order to justify that our algorithm performs better than current existing context aware algorithms.

The rest of the paper is organized as follows. Section II, explains the related work in this field. Novel kernel-context recommendation algorithm is described in Section III and all its variants are described in Section IV. The experimental setup is explained briefly in Section V. In Section VI results are presented in detail followed by the conclusion in Section VII.

C. PROBLEM STATEMENT AND DESIGN OBJECTIVES

At present, context-aware recommender systems are considered as a new entrant and relatively little research work has been done in this domain. Their importance; however, has started establishing. The computing world has realized that in certain scenarios the accuracy of recommendations is extremely dependent on the context and old techniques and methods cannot satisfy user's needs and interests according to the current context. A few researchers have anticipated the importance of context in recommender systems for improvement in the accuracy of predictions. Existing algorithms and approaches deal with the 2D user-item matrix and use these 2D ratings for further predictions and suggestions to a new user. These methods provide good recommendations but still not precise to the user's needs.

In context-aware recommender systems, the selection of context is also an important issue. In context-aware recommender systems, a variety of contextual information can be considered. Some of them are important and others can be less important in a recommendation process at a given time scale or in a particular scenario. So, context selection is also an important issue and hence, care and consideration should be given to the selection process.

The proposed work aims to improve the quality of recommendations by investigating various well-known literature techniques of adding context in recommender systems. Against the aforementioned problem statement, this paper aims to develop a Kernel based context-aware recommender system algorithm and compare its performance with the existing approaches (Kernel Mapping Recommender System (KMR)) and Post-Filtering concept - the most preferable concept in the literature of context aware RS.

Different sources of information are used for making recommendations or predictions. Our proposed framework is flexible enough to allow these sources to be incorporated, by combining different kernels based on the vectors of various information sources. In particular, this information in our work is contextual information of items and users that are combined through point-wise multiplication as well as concatenated using additive model.

II. RELATED WORK

Nowadays, context-aware recommender systems are taken as a new topic and relatively less work has been done in this domain. Former approaches and methods cannot accomplish the user's needs and interests according to the current context. A number of researches [19]–[21] have been carried out in this domain, which proposes various techniques and methods that use context for the recommendation. In this domain, we are dealing with different types of datasets in the field of recommender systems. They can be of movies, music, books, articles etc. In the field of music, Schedl *et al.* [22] included the user's context as social context, time of day, weather, mood and various other factors.

A context is a vast term that can be defined in various ways in different situations. The user's social environment like social friends and social interaction with others can be considered as a user's social context [23]. A variety of applications and systems are available that use context for movies and songs recommendations e.g. Smart Radio [24]. Such systems use the history of the user or the playlist that a user is currently listening to, as a current context. Baltrunas [25] describe a general architecture of context-aware recommender systems.

Researchers [26]–[29] have used time as context and discussed various factors and the effects of using time context in recommender systems. Various context-aware recommender systems use Collaborative Filtering (CF) techniques for incorporating context because it is one of the most popular recommendation techniques. An approach described in [26] followed the collaborative filtering techniques on sliced data corresponding to the current context. In order to model the time context, tensor factorization can also be used reference. Some researchers [30] explained the modeling of different contexts instead of any specific one. There are various approaches that utilize a user context extracted by different sources [31]–[33]. Other information like location, weather, and mood can also be used as user context.

In [34]–[38], researchers verified the importance of contextual information in the domain of recommender system in the process of recommendations. Traditional CF algorithms can make recommendations only by considering static states of user and item. In fact, user preferences change over time and cannot be measured by the traditional systems, which is the major drawback of all the existing state of the art algorithms.

Linden *et al.* [39] propose context aware predictor based on factorization machines [17]. Factorization machines (FMs) simulates the most successful approaches in the recommender systems such as SVD++ [40], matrix factorization [41], PITF [42] etc.

In most of the previous approaches, the context is integrated with the collaborative filtering techniques and build a 3D concept in CF. Ghazanfar *et al.* [14] have introduced Kernel Mapping Recommender Systems (KMR Algorithm) that provides accurate recommendations and has state-of-the-art performance than other existing CF recommender systems. Ghazanfar and Prvgel-Bennett [43] proposed a structure that describes how to add contextual data into KMR algorithms to further increase its precision and accuracy factor.

In [43] the rating function "f" for the Items "I" rated by Users "ţ" for the two-dimensional user-item matrix can be represented as:

$$f : U \times I \rightarrow R$$

but in the case of contextual information, context C is associated with every rating. The rating function "f" for this is represented as:

$$f : U \times I \times C \rightarrow R \tag{1}$$

$$f : (U^{c1} \times I^{c1} \times C_1) + (U^{c1} \times I^{c2} \times C_2) + \ldots\ldots (U^{c_c} \times I^{c_c} \times C_c) \rightarrow R \tag{2}$$

Ghazanfar and Prvgel-Bennett [43] deliberated the social context concept and took friends of an active user in a social context. They shed light on the improvement of an accuracy factor in recommendations by incorporating contextual information in the forms of friends, friends of friends (FOAF) and so on. Linas *et al.* [44] proposed a Context Aware Matrix Factorization Algorithm (CAMF), which utilizes the interaction between users and contexts. In most of the work it is used as a baseline model for context aware algorithms and models. In [21] Differential context modeling (DCM) Approach has explained, in which the 'differential' part divides the algorithm into different functional parts upon which contextual constraints are applied. Authors proposed two versions of this technique that is differential context relaxation (DCR) and differential context weighting (DCW). In [20], researchers have provided a splitting concept in context aware systems based on item-splitting (pre-filtering) and user-splitting (pre-filtering). As a result, the authors proposed a hybrid of User and item splitting as UI Splitting while considering contextual information of both users and items. Moreover, Bias TF-RT Model in [45] is an integrated approach of Bias Tendor Factorization model and Context Feature Auto-ending Algorithm.

The disparity of a huge literature shows that only a little work has done in this field of research. Although certain basic approaches of *pre-filtering* and *post-filtering* of contextual information are being used; however, these approaches cannot be considered as standard techniques because they are figured as context unaware recommender systems [39]. These approaches either pre-process or post-process the given data based on the context of an active user's interest before (or after in case of post-filtering) applying the recommendation algorithm [46]. All such former approaches do not support the way in which the contextual information has been incorporated during recommendation. Hence, to overcome such problems where user's opinion would be very important, and prediction should be according to their current needs, we propose a unified framework that not only focuses on the user's current needs by considering user's current contextual information but also increases the overall performance of the system.

III. PROPOSED KERNEL CONTEXT RECOMMENDER SYSTEM ALGORITHM

In this section, we describe our context-aware recommender system namely KCR. The core notation and idea behind such systems are to take the contextual information into account about the item and the user while making recommendations, which makes the recommendations more relevant and accurate according to the current context and user's needs.

A. KCR ALGORITHM: EXPLOITING CONTEXT IN KERNEL MAPPING RECOMMENDER SYSTEM

We have proposed a context-aware framework for both user- and item-based versions by using the kernel mapping concept in collaborative filtering. The Kernel mapping recommender system algorithm is based on a novel structure learning technique. This framework has the flexibility to exploit various user and item related contexts during the recommendation process using different kernels that influence the performance of the system by improving the predictive accuracy, scalability and flexibility factors. The Basic framework of the proposed Kernel Context Recommender (KCR) algorithm is presented in Figure 1. This Figure explains the overall graphical structure of our framework that how it works and what kind of information it considered while providing recommendations. Our proposed model, work on the 3D concept of adding contextual information along with the rating matrix

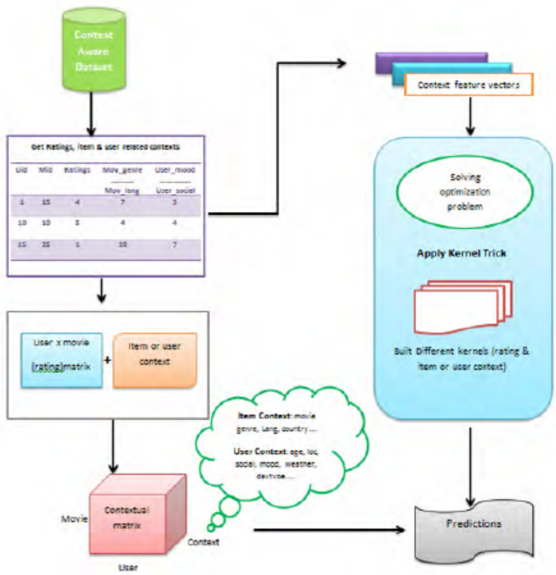

FIGURE 1. Graphical representation of proposed context aware kernel mapping recommender system (KCR) algorithm.

and generate predictions by combining them rather than using context before or after predicting rating. Moreover, we did this by introducing different feature vectors for different contexts. We considered the initial work done in [43] and [14]. The KMR Algorithm provides the state-of-the-art performance and resolves various problems in RS; however, it cannot deal with contextual information. In order to eliminate this issue, we extended it by introducing various kernels in KMR algorithm. Ghazanfar and Prvgel-Bennett [43] introduced the concept of adding simple context (i.e. social context (foaf)) in KMR and a very small experiment was conducted to explain the addition of contextual information. In our work, we introduced the additive and multiplicative models for both item- and user-based versions of our Context Aware Algorithm and proposed a framework for unifying more contextual information. We also generate the post-filtering model as an extension of our algorithm in order to compare the results for all versions and models of the algorithm and empirically show that our framework is better than the existing techniques.

B. BASIC KERNEL MAPPING RECOMMENDER SYSTEM ALGORITHM

We are using kernel mapping method concept that gives state-of-the-art performance for solving the recommendation problem. The idea behind these methods is to find a multilinear mapping between two high dimensional vector spaces. This mapping is learned to solve the quadratic optimization problem. The solution to this problem is to find the inner product of two vector spaces. We follow the idea developed by Joachims [47] that trains the system in linear time. The KMR algorithm is based on the structured learning algorithm and Szedmak *et al.* [48] proposed an algorithm for learning incomplete data sets. Next, we briefly describe how KMR solves the recommendation problem effectively using a structured learning approach.

1) ITEM-BASED KMR

In the basic KMR algorithm, we are dealing with the actual ratings given by users and all other ratings are assumed as missing values:

$$r_{iu} = \begin{cases} r_{iu} & \text{if } r_{iu} \text{ is given,} \\ \emptyset & \text{Otherwise (Missing value).} \end{cases}$$

Additive and multiplicative models are built for residual ratings in order to perform the recommendation task. The additive model of residual ratings is shown in the equation given below:

$$\hat{r}_{iu} = r_{iu} - \bar{r}_i - \bar{r}_u + \bar{r}. \tag{3}$$

where $\bar{r}_i$ is a mean rating for items, $\bar{r}_u$ is a mean rating for users, $\bar{r}$ is the overall mean rating. The multiplicative model for residual ratings can be stated as:

$$\hat{r}_{iu} = \frac{r_{iu}\tilde{r}}{\tilde{r}_i\tilde{r}_u}. \tag{4}$$

where $\tilde{r}_i$, $\tilde{r}_u$ and $\tilde{r}$ are the geometric mean of the ratings given to item i, the ratings of user u and for the whole ratings. Here, we are working on both additive and multiplicative model, but we found better results on a multiplicative model.

In order to learn structured data, we use a technique proposed by Szedmak and his co-workers [48]. This developed approach is further modified to resolve various collaborative filtering problems. We have assumed that some item-related information is represented as q_i. Assume, this information could be the ratings r_{iu} (where $r_{iu} \mid (i, u) \in \mathcal{D}$ and $\mathcal{D} \subset \mathcal{I} \times \mathcal{U}$ is the set of user-item pairs that have been rated.) for $u \in \mathcal{D}_i$ ($\mathcal{D}_i$ is the set of ratings of the users who have rated an item i) or some textual features that describes an item i. The input features, q_i, are mapped by a function ϕ into a Hilbert space. Similarly in some other Hilbert space the residual ratings i.e. $\hat{r}_{iu}$ are also mapped to some vector. All such objects reside in a function space of $L_2(\mathbb{R})$ in this work. Using mean $\hat{r}_{iu}$ and variance σ, each residual is represented by the density function of a normal distribution as follows:

$$\psi(\hat{r}_{iu}) = \mathcal{N}(x|\hat{r}_{iu}, \sigma). \tag{5}$$

While assigning a rating, the incentive of choosing this concept is to model out the rating errors, which can either be generated due to discretization of the rating scale or variability in providing rating (under certain context, e.g. due to change of mood on the day user has made rating or change of social environment etc.). We seek the mapping between two vector spaces e.g. input vector space and other space is based on residual ratings vector which is useful in making predictions. More precisely, in our work we are looking for a set of linear mappings, $\mathbf{W}_u$, from the Hilbert space containing the vectors $\phi(q_i)$ representing the contextual features into the Hilbert space of the representation of the residuals, $\psi(\hat{r}_{iu})$.

To this end we minimize the Frobenius norm of $\mathbf{W}_u$ and the sum of slack variables, ξ_i, with respect to a set of maximum margin type constraints in the following optimization problem:

$$\begin{aligned} &\min \ \frac{1}{2}\sum_{u\in\mathcal{U}}\|\mathbf{W}_u\|^2 + C\sum_{i\in\mathcal{I}}\xi_i \\ &\text{with respect to } \mathbf{W}_u,\ u\in\mathcal{U},\ \xi_i,\ i\in\mathcal{I} \\ &\text{subject to } \left(\psi(\hat{r}_{iu}), \mathbf{W}_u\phi(q_i)\right) \geq 1-\xi_i \end{aligned} \tag{6}$$

The optimum is achieved when both vectors $\mathbf{W}_u\phi(q_i)$ and $\psi(\hat{r}_{iu})$ become align uniformly. After learning the mappings of $\mathbf{W}_u$, we can then make predictions for a new item (k) by using $\mathbf{W}_u\phi(q_k)$. In order to solve the optimization, the problem we introduce Lagrangian multipliers:

$$\begin{aligned} \mathcal{L} = \frac{1}{2}\sum_{u\in\mathcal{U}}\|\mathbf{W}_u\|^2 + C\sum_{i\in\mathcal{I}}\xi_i - \sum_{(i,u)\in\mathcal{D}}\alpha_{iu} \\ \times\left(\left(\psi(\hat{r}_{iu}), \mathbf{W}_u\phi(q_i)\right) - 1 + \xi_i\right) - \sum_{i\in\mathcal{I}}\lambda_i\xi_i \end{aligned} \tag{7}$$

where, $\alpha_{iu} \geq 0$ is a Lagrangian multiplier introduced to ensure that the term $\left(\psi(\hat{r}_{iu}), \mathbf{W}_u\phi(q_i)\right) \geq 1-\xi_i$ and $\lambda_i \geq 0$ are another Lagrangian multipliers, which is used to ensure $\xi_i \geq 0$. The optimal results of mapping are found by performing:

$$\min_{\{\mathbf{W}_u}, \xi_i\} \max_{\{\alpha_{iu}}, \lambda_i\}\mathcal{L},$$

Subjected to the constraints $\alpha_{iu} \geq 0$ for all $(i, u) \in \mathcal{D}$ and $\lambda_i \geq 0$ for all $i \in \mathcal{I}$. For a general linear mapping of $\mathbf{W}_u$, we have

$$\frac{\partial}{\partial \mathbf{W}_u}\left(\psi(\hat{r}_{iu}), \mathbf{W}_u\phi(q_i)\right) = \psi(\hat{r}_{iu}) \otimes \phi(q_i),$$

where '$\otimes$' represents the tensor product of two vectors. In case of linear mappings, mapping of $\mathbf{W}_u$ can be signified by a matrix as Hilbert or high dimensions vector space has finite dimensions. If we take the derivative of equation (7) with respect to $\mathbf{W}_u$, then we get:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{W}_u} = \mathbf{W}_u - \sum_{i\in\mathcal{D}_u}\alpha_{iu}\psi(\hat{r}_{iu}) \otimes \phi(q_i).$$

The above given expression describes that the Lagrangian variables are minimized here with respect to $\mathbf{W}_u$. Now,

$$\mathbf{W}_u = \sum_{i\in\mathcal{D}_u}\alpha_{iu}\psi(\hat{r}_{iu}) \otimes \phi(q_i)$$

If we take the derivative of the aforementioned equation (7), then:

$$\frac{\partial\mathcal{L}}{\partial\xi_i} = C - \sum_{u\in\mathcal{D}_i}\alpha_{iu} - \lambda_i$$

We find that the Lagrangian multipliers are minimized w.r.t. ξ_i, when these derivatives are put as 0. This is true when,

$$\sum_{u\in\mathcal{D}_i}\alpha_{iu} = C - \lambda_i \leq C, \quad \textit{where} \quad \lambda_i \geq 0$$

When we substitute the values of all sub-expressions into the Lagrangian equation (7), a dual problem of equation (6) is obtained which is a maximization issue w.r.t. the Lagrangian variable α_{iu}. The dual problem in mathematical form is expressed as:

$$\begin{aligned} f(\alpha) = -\frac{1}{2}\sum_{u\in\mathcal{U}}\sum_{i,i'\in\mathcal{D}_u}\alpha_{iu}\alpha_{i'u}\left(\psi(\hat{r}_{iu}), \psi(\hat{r}_{i'u})\right) \\ (\phi(q_i), \phi(q_{i'})) + \sum_{(i,u)\in\mathcal{D}}\alpha_{iu} \end{aligned}$$

subject to the constraint on α i.e. $\alpha \in Z(\alpha)$ Where,

$$Z(\alpha) = \left\{\alpha \middle| \forall u \in \mathcal{U},\ \sum_{u\in\mathcal{D}_i}\alpha_{iu} \leq C \ \wedge\ \forall (i,u) \in \mathcal{D},\ \alpha_{iu} \geq 0\right\}.$$

We can now use a Kernel Trick here. We apply different kernel functions which can be defined as:

$$\begin{aligned} K_{\hat{r}}(\hat{r}_{iu}, \hat{r}_{i'u}) &= \left(\psi(\hat{r}_{iu}), \psi(\hat{r}_{i'u})\right) \\ K_q(q_i, q_{i'}) &= (\phi(q_i), \phi(q_{i'})) \end{aligned}$$

By using these two kernels we can represent $f(\alpha)$ as:

$$\begin{aligned} f(\alpha) = -\frac{1}{2}\sum_{u\in\mathcal{U}}\sum_{i,i'\in\mathcal{D}_u}\alpha_{iu}\alpha_{i'u}K_{\hat{r}}(\hat{r}_{iu}, \hat{r}_{i'u})K_q(q_i, q_{i'}) \\ + \sum_{(i,u)\in\mathcal{D}}\alpha_{iu} \end{aligned}$$

Kernel functions can be of different types and we can choose any of the positive definite kernel functions. By applying kernels, we computed that the mapping of residual ratings through Normalization as discussed before becomes inexpensive.

2) LEARNING THE LAGRANGE MULTIPLIERS

Solving the quadratic programming problem using a general or universal quadratic programming solver in large scale RS would be very expensive and cause practicality issues due to a large number of data points, users or items. For this reason, we find the Conditional Gradient Method as an appropriate alternative to this problem. For understanding the concept of Conditional Gradient Method, it is better to represent $f(\alpha)$ in a matrix form. Further details about learning the Lagrangian multipliers are extracted using the methodology described by Ghazanfar *et al.* [14], in which a complete derivation is described.

Linear programming problem can solve the problem in a linear time. The complexity, for all the users, is $O(|D|)$ which concludes the complexity of algorithm as linear.

C. INCORPORATING CONTEXT IN ITEM BASED KCR ALGORITHM

KMR algorithm provides predictions by using kernel trick and by building kernels against the given information related to input features and residual ratings. The KMR algorithm provides predictions by using a 2D concept i.e. it considers only two-dimensional user-item rating matrix.

Although KMR algorithm has dominant results in accuracy and increases the system performance, according to the increasing demand of adding context and by knowing the importance of context in the prediction model, we have proposed a Context Aware KMR framework by using the flexibility of this algorithm.

In the item-based version of this framework, we need to add user-related context. In this version, basic KMR algorithm uses various users that have rated a particular movie.. Contextual kernels are built separately using users' related information e.g. time, day type, season, location, weather, social context, mood, dominant emotions, decision and physical health. Then these kernels are combined along with the residual rating kernel so that this information can also be utilized while making a prediction.

$$K_{\hat{r}}(\hat{r}_{iu}, \hat{r}_{i'u}) = \left(\psi(\hat{r}_{iu}), \psi(\hat{r}_{i'u})\right) \rightarrow ResidualRatingKernel.$$

Similarly, we can define context kernel formally:

$$K_{context}(contextvector_{iu}, contextvector_{i'u}) \rightarrow ContextualKernel.$$

where '*context*' can be time, day type, season, location, weather, social context, mood, dominant emotions, decision and physical health of a user.

D. DEFINING DIFFERENT CONTEXT KERNELS

In this work, we have used 9 types of user related contextual information. The description of various users' context kernels that are used in this work is discussed as follows:

1) **Time Kernel**, denoted by $\mathbf{K}_{time}$, uses the time feature vector. It elaborates the time context e.g. morning, afternoon, evening, the night of a user in which he has rated that particular item.

$$K_{time}(t_{iu}, \hat{t}_{i'u}) \rightarrow TimeKernel.$$

2) **Daytype Kernel**, denoted by $\mathbf{K}_{daytype}$, uses the day-type feature vector. It describes the context of a day in which the user has rated an item e.g. it's a working day, weekend or holiday.

$$K_{daytype}(d_{iu}, \hat{d}_{i'u}) \rightarrow DayTypeKernel.$$

3) **Season kernel**, denoted by $\mathbf{K}_{season}$, considers a season vector and the values assigned to this type of context are summer, winter, autumn, spring.

$$K_{season}(s_{iu}, \hat{s}_{i'u}) \rightarrow SeasonKernel.$$

4) **Location kernel**, denoted by $\mathbf{K}_{loc}$, considers the location of a user from where a user had rated item or items. Here location doesn't mean the global location but here it is about the home, office, public place, friend's house.

$$K_{loc}(l_{iu}, \hat{l}_{i'u}) \rightarrow LoactionKernel.$$

5) **Weather kernel**, denoted by $\mathbf{K}_{weather}$, describes the weather conditions of a user while rating items e.g. weather conditions will be sunny day/clear, snowy, rainy, stormy, cloudy when the user has rated a particular item.

$$K_{weather}(w_{iu}, \hat{w}_{i'u}) \rightarrow WeatherKernel.$$

6) **Social context**, denoted by $\mathbf{K}_{social}$, is built on the type of social circle with which a user was rating an item e.g. alone, partner, friends, colleagues, parents, my family, public.

$$K_{social}(s_{iu}, \hat{s}_{i'u}) \rightarrow SocialContextKernel.$$

7) **Emotion kernel**, denoted by $\mathbf{K}_{emo}$, is based on the feature vector of emotions of a user. The values on which emotion vector is made of are sad, happy, scared, surprised, angry, disgusted, and neutral.

$$K_{emo}(e_{iu}, \hat{e}_{i'u}) \rightarrow EmotionsKernel.$$

8) **Mood kernel**, denoted by $\mathbf{K}_{mood}$, is based on the mood of a user e.g. a user may be in a positive mood, can be an n negative mood or neutral.

$$K_{mood}(m_{iu}, \hat{m}_{i'u}) \rightarrow MoodKernel.$$

9) **Decision kernel**, denoted by $\mathbf{K}_{decision}$, is based on the **decision** of a user that whether a user has his own decision of watching that movie or someone has suggested them.

$$K_{dec}(d_{iu}, \hat{d}_{i'u}) \rightarrow DecisionKernel.$$

10) **User's health kernel**, denoted by $\mathbf{K}_{health}$, is based on the basis of an information related to the physical health of a user i.e. whether a user has ranked an item in healthy condition or an illness.

$$K_{health}(h_{iu}, \hat{h}_{i'u}) \rightarrow User'shealthKernel.$$

E. COMBINING VARIOUS CONTEXTUAL KERNELS

For making predictions (and recommendation) there are numerous sources of information that can be utilized. The defined sources can be accommodated by combining the aforementioned kernels. We can make our predictions more accurate and make the system performance better by including context.

These Kernels are combined *linearly*.

$$K = K_{rat} + K_{time} + K_{season} + K_{loc} + K_{weathr} + K_{social} + K_{emo} + K_{mood} + K_{dec} + K_{health}. \quad (8)$$

In Equation 8, these kernels have the same contribution in the prediction model. It can also be tuned by introducing a factor which exhibits the level of contribution that a particular kernel contributes. These parameters are tuned by computing and analyzing its generalization performance over the training set.

$$\begin{aligned} K = {} & \beta_{rat}K_{rat} + \beta_{time}K_{time} + \beta_{season}K_{season} + \beta_{loc}K_{loc} \\ & +\beta_{weathr}K_{weathr} + \beta_{social}K_{social} + \beta_{emo}K_{emo} \\ & +\beta_{mood}K_{mood} + \beta_{dec}K_{dec} + \beta_{health}K_{health}. \end{aligned} \quad (9)$$

The kernel K is a convex combination of the contextual kernels.

We assume that $\beta_{rat} + \beta_{time} + \beta_{season} + \beta_{loc} + \beta_{weathr} + \beta_{social} + \beta_{emo} + \beta_{mood} + \beta_{dec} + \beta_{health} = 1$ without the loss of generalization. So, we tune these parameters from the range of 0.0 to 1.0.

Combining kernels in this way actually means that we are dealing with the vectors that belong to these contexts which can be represented as:

$$\phi_{context} = \phi_{rat} + \phi_{time} + \phi_{season} + \phi_{loc} + \phi_{weathr} + \phi_{social} + \phi_{emo} + \phi_{mood} + \phi_{dec} + \phi_{health}. \quad (10)$$

These Kernels are combined linearly and non-linearly both for additive and multiplicative models.

1) ADDITIVE MODEL

The way in which kernels are combined in equation (8) and (10) can be concatenated in the additive model.

$$\phi_{context} = \phi_{rat} \oplus \phi_{time} \oplus \phi_{season} \oplus \phi_{loc} \oplus \phi_{weathr} \oplus \phi_{social} \oplus \phi_{emo} \oplus \phi_{mood} \oplus \phi_{dec} \oplus \phi_{health}. \quad (11)$$

where '$\oplus$' represents the direct sum of feature vectors.

2) MULTIPLICATIVE MODEL

Alternatively, these vectors or kernels can also be combined in *non-linearly*. In this type of model

$$K = K_{rat} \cdot K_{time} \cdot K_{season} \cdot K_{loc} \cdot K_{weathr} \cdot K_{social} \cdot K_{emo} \cdot K_{mood} \cdot K_{dec} \cdot K_{health}. \quad (12)$$

where '·' represent the point-wise product of these kernel matrices. Similarly, feature vectors are multiplied in case of a multiplicative model.

$$\phi_{context} = \phi_{rat} \otimes \phi_{time} \otimes \phi_{season} \otimes \phi_{loc} \otimes \phi_{weathr} \otimes \phi_{social} \otimes \phi_{emo} \otimes \phi_{mood} \otimes \phi_{dec} \otimes \phi_{health}. \quad (13)$$

where '$\otimes$' represents the tensor product of contextual feature vectors.

IV. EXTENSION TO THE BASIC ALGORITHM

Here, we have explained some of the extensions to the basic algorithm.

A. USER BASED VERSION OF KCR ALGORITHM

Different models can be trained along the columns or rows of the data matrix depending upon the data-set features and characteristics; for example, the number of users that have rated a particular item, number of items rated by an active user etc. So, various models or algorithms have been proposed that deals with these characteristics. A related algorithm is proposed by extending our proposed work from the user's point of view. Hence, this extension of our proposed model is named as *User-Based KCR Algorithm*.

For User-Based *KCR* Algorithm, we are performing user-based predictions and recommendations. In this version, we use information q_u about the users u and in order to align various feature vectors q_u to residual ratings ($\hat{r}_{iu}$), linear mapping $\mathcal{W}_i$ is performed. The derivations for user based version is similar to that of the item based recommender system but the subscripts i and u are interchanged. The context that is incorporated in this version is items' related context because in this case we are concerned with movies and its context rated by a particular user. In the LDOS-CoMoDa dataset, item related contextual information that we are using are: movie genre, country, year of release, language, and movie director. All the kernels built in the training model are based on these context variables or information.

B. DEFINING DIFFERENT CONTEXT KERNELS

In this work, we are concerned with 5 types of item related contextual information for LDOS-CoMoDa dataset.

1) **Genre kernel**, denoted by $\mathbf{K}_{genre}$, describes the genre of a movie that a movie belongs to which type of genre e.g. Horror, Thriller, Action, Romantic, Sci-fi, Comedy, Drama etc.
2) **Movie Country Kernel**, denoted by $\mathbf{K}_{country}$, describes the country of a movie to which that movie belongs.
3) **Movie Release Year Kernel**, denoted by $\mathbf{K}_{year}$, defines the year in which the movie was released.
4) **Movie Director kernel**, denoted by $\mathbf{K}_{fea}$, is based on a director who has directed that movie.
5) **Movie Language kernel**, denoted by $\mathbf{K}_{lan}$, is based on the language of a movie whether it is in English, French, Chinese, Korean etc.

These Kernels can also be combined linearly and non-linearly both for additive and multiplicative models similar to the item-based version.

V. EXPERIMENTAL SETUP

A. DATASET

Most of the datasets used in the recommender system's literature are movie datasets. Most of these datasets do not have any contextual information mainly about users. So, we need to crawl a context-aware dataset by collaborating with various companies and other online service providers in a proper way. The datasets used in this research work are described as below:

The **LDOS-CoMoDa** is the one of the most widely used movie dataset that uses user's context. This dataset has twelve potentially relevant contextual variables for movie selection. This dataset is also a movie dataset which acquires 268 users, 4381 movies and 2278 rating records having rating criteria from 1 to 5, where 1 is considered as the worst and 5 as the best. The distribution of ratings over the whole dataset is described in Figure 2. The difference between two consecutive rating scales is 1. This dataset has a lot of contextual information regarding users and movies. Contextual variables or contextual information given in this dataset are: Time (Morning, Afternoon, Evening, Night), Day-type (Working day, Weekend, Holiday), Season (Spring, Summer, Autumn, Winter), location (Home, Public place, Friend's house), Weather (Sunny/clear,

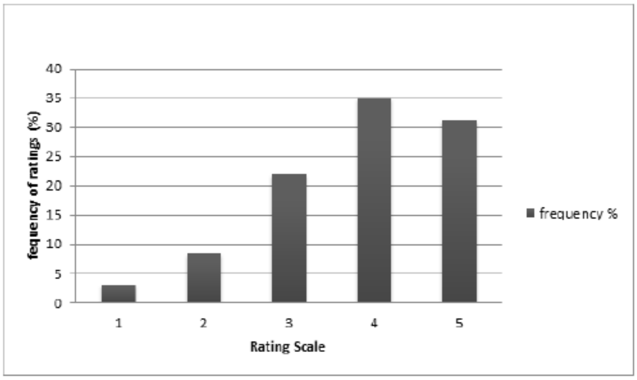

FIGURE 2. Rating distribution for LDOS-CoMoDa dataset.

Rainy, Stormy, Snowy, Cloudy), Social (Alone, My partner, Friends, Colleagues, Parents, Public, My family), EndEmo (Sad, Happy, Scared, Surprised, Angry, Disgusted, Neutral), DominantEmo (Sad, Happy, Scared, Surprised, Angry, Disgusted, Neutral), Mood (Positive, Neutral, Negative), Physical (Healthy, ill), Decision (User decided which movie to watch, User was given a movie), Interaction (first interaction with a movie, n-th interaction with a movie). It also has the context of movies like the language of movie, movie country, year of release, genre, director etc. The contextual information along with their types of feature vectors describes in Table 2.

The **DePaulMovie** dataset is movies-based dataset with some context information. It contains three contexts, which are used for rating different movies. This dataset was crawled by a survey at DePaul University and collected by one of the researchers working on it. Students rated various movies in different contexts such as: location, time, companions. Variation in contexts causes variation in ratings for movies. This dataset contains 97 users, 79 Movies and 2720 rating records. Rating criteria for this dataset are from 1 to 5, where 1 is the worst and 5 is the best. The distribution of ratings over the whole dataset is shown in Figure 3. User-related contextual variables (containing different information such as, time (weekend, weekday), location (Cinema, Home) and companion (Alone, Partner, Family)) are shown in Table 3.

The summarized characteristics of datasets used in this research are given in Table 4.

B. METRICS

The objective of this research is to provide efficient recommendations using the item ratings provided by users and to evaluate the performance of the recommendation system. Predictive accuracy metrics and classification accuracy metrics have been used to evaluate the performance of the recommendation systems. These evaluation metrics are further used to provide benchmark and to compare the performance of the different recommender systems.

- **Root Mean Square Error (RMSE)** RMSE is one of the most widely used evaluation parameter for recommender systems and closely relate to Mean absolute error and squared the error before summing it. The purpose is to emphasize on larger errors. For example, an error at one point may increase the sum by one

TABLE 2. Detail about the User's contextual information and context feature vectors of LDOS-CoMoDa dataset.

Context	Context Feature Information
Time	Morning Afternoon Evening Night
Day-Type	Working day Weekend Holiday
Season	Spring Summer Autumn Winter
location	Home Public place Friend's house
Weather	Sunny / clear Rainy Stormy Snowy Cloudy
Social	Alone My partner Friends Colleagues Parents Public My family
EndEmo	Sad Happy Scared Surprised Angry Disgusted Neutral
DominantEmo	Sad Happy Scared Surprised Angry Disgusted Neutral
Physical	Healthy ill
Decision	User decided which movie to watch User was given a movie
Interaction	first interaction with a movie n-th interaction with a movie

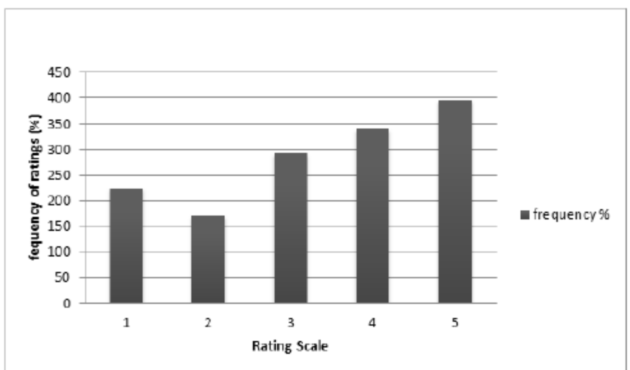

FIGURE 3. Rating distribution for DePaulMovie dataset.

TABLE 3. Detail about the user's contextual information and context feature vectors of DePaulMovie dataset.

Context	Types of context feature vectors
Time	Weekend Weekday
location	Home Cinema
Companion	Alone Partner Family

and if an error occurs at both points then the sum will be increased by four. The aim of an algorithm is to minimize the RMSE score. The formula for calculating RMSE is:

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{D}_{test}|}\sum_{r_{iu}=1}^{|\mathcal{D}_{test}|}(r'_{iu} - r_{iu})^2}$$

where, r'_{iu} are predictive ratings, r_{iu} are actual ratings and $|\mathcal{D}^{test}|$ are the test set records.

- **F1-Measure** The parameter F1 measure is the evaluation metric used to measure the effectiveness of recommender system. It can be measured by analyzing the frequency with which it helps users to predict or recommend good items.
  The probability of relevant items from all the items recommended by the system is termed as precision. Mathematically, precision can be defined as:
  $$\text{Precision} = \frac{Items_{relevant\_selected}}{Items_{total\_selected}},$$
  The probability of selecting relevant items from the total number of relevant items is termed as recall. It can also be defined in mathematical form as:
  $$\text{Recall} = \frac{Items_{relevant\_selected}}{Items_{total\_relevent}},$$
  The F1 measure can be calculated by using the previous two metrics, i.e. precision and recall. In mathematical form, it is computed as:
  $$\text{F1} = \frac{2 \times precision \times recall}{precision + recall}$$

C. EVALUATION METHODOLOGY

The proposed work use the dataset by randomly dividing the dataset into training and testing sets using 5-fold cross validation. More Specifically, the movies dataset is used and we randomly divide 80% movies as training set and the rest (20%) as the test set. To train the parameters, the training set id further divided into training and validation sets using 80-20 rule. Then we have calculated the residual ratings required for the rating vector.

VI. RESULTS AND DISCUSSION

In this section, we describe the detailed results and comparisons of our experiments. The main variants of our proposed algorithm are context aware item-based (KCR_{ib}) and context-aware user-based (KCR_{ub}) version. For different combinations of information, for example, in the case of item-based rating and context we use $KMR_{ib\oplus u_{context}}$ and $KMR_{ub\oplus i_{context}}$ in the case of user based rating plus user context, when we are adding different kernels. When we are concerned about multiplying various kernels then these cases are denoted as: $KMR_{ib\otimes u_{context}}$ and $KMR_{ub\otimes i_{context}}$, respectively. We have also implemented the post-filtering concept and then perform result comparison as it is a weel used concept in most of the research articles.

These comparisons are based on the variations among different evaluation parameters for the whole dataset (along with some missing contextual information) and for the denser dataset (consider only those users who have all contextual information). We have also compared our proposed model (KCR) results with the most preferable technique of context-aware recommender system in literature i.e. post-filtering and other work done in this domain like Context Aware Matrix Factorization Algorithm (CAMF) [44], Differential context modeling (DCM) Approach in [21] and a context aware approach proposed in [20]. Similarly, [49] proposes a context aware algorithm (CSVD), which is the combination of Singular Value Decomposition (SVD) algorithm and post context filtering concept.

In this thesis, we are focusing on evaluation metrics including Root Mean Square Error (RMSE) and F1-measure.

A. EVALUATING PERFORMANCE ON WHOLE DATASET IN TERMS OF EVALUATION PARAMETERS

The algorithm is tested on the whole dataset with all users, with or without the contextual information (it means dataset

TABLE 4. Characteristics of datasets used in this work.

Characteristics	Dataset	
	(LDOS-CoMoDa)	(DepaulMovie)
Number of Users	121	97
Number of Movies	1232	79
Number of Ratings	2278	2720
Rating Scale	1 (bad)-5 (excellent)	1 (bad)-5 (excellent)
Sparsity	0.988	0.188
Contextual Information	Gender, daytype, time, social, mood, emotions, location, country, weather, decision, movie genre, language, country, year of release, director.	Time, Location, Companion

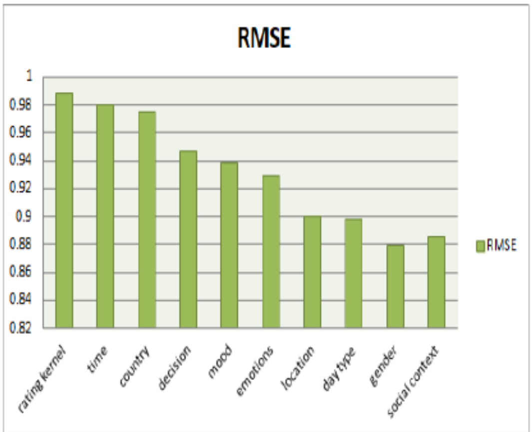

FIGURE 4. Results of adding various user context kernels in additive KCR. X-axis = different context kernels, Y- axis = RMSE, Dataset= LDOS-CoMoDa (full dataset) model = additive KCR, version = item-based.

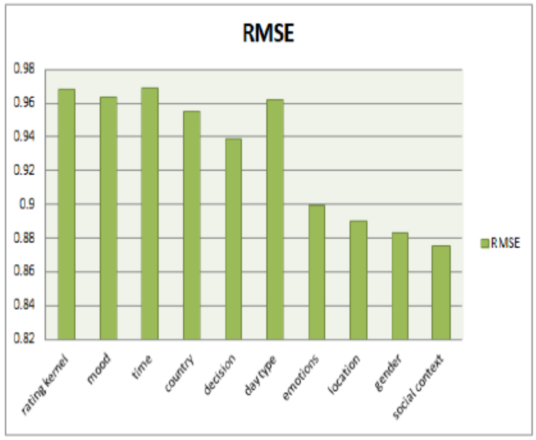

FIGURE 5. Results of different user context kernels in multiplicative KCR. X-axis = different context kernels, Y- axis = RMSE, dataset= LDOS-CoMoDa (full dataset) model = multiplicative KCR, version = item-based.

is sparser). The results of different proposed models are discussed below:

1) ITEM-BASED VERSION

This version has contextual information that is based on user context as in (item-based) IB version we are finding the number of users for a particular item. So, the contexts used here are user country, location, age, gender, time, mood, emotions, day-type and social context of the user. We have used various types of the kernel; such as, polynomial, Gaussian, Poly-Gaussian, Poly-Laplace, etc. based on the normalization type. In this work, we found good results by considering rating kernel as the polynomial kernel and the additional contextual kernels as the poly-Gaussian kernels. The graphs given below describe the variations in various evaluation parameters due to the concatenation of different kernels built on different contexts.

1) ***Additive Model***: When kernels are added or concatenated linearly, then the results of including various contexts for item-based version ($KCR_{ib\oplus}$) are shown in Figure 4. Figure 4 shows that the RMSE decreases when we add various contexts. The percentage decrease in RMSE for various contexts is 0.015% for decision, 0.04% for mood, 0.05% for emotions, 0.131% for location, 0.146% for daytype, 0.156% for gender and 0.191% for social context, respectively.
2) ***Multiplicative Model***: When kernels are non-linear and are multiplied point wise then the results of including various contexts for item-based version i.e. $KCR_{ib\otimes}$ are shown in Figure 5.

2) USER-BASED VERSION

Here we are considering different contextual information about movies. In this work, we found good results by considering rating kernel as the polynomial kernel and the additional contextual kernels as Poly-Gaussian kernel. The graphs given below discuss the variation of various parameters.

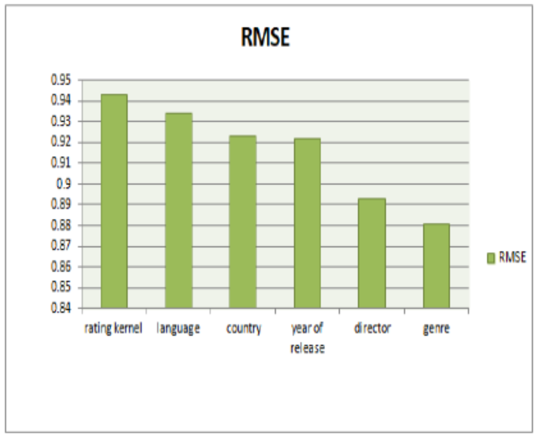

FIGURE 6. Results of different item context kernels in additive KCR. X-axis = different context kernels, Y- axis = RMSE, dataset= LDOS-CoMoDa (full dataset) model = additive KCR, version = user-based.

1) ***Additive Model***: When kernels are added or concatenated linearly then the results of including various contexts for user-based version ($KCR_{ub\oplus}$) are shown in Figure 6. This figure shows that the RMSE decreases when we add various contexts in user-based KMR. The percentage decrease in RMSE for various contexts is 0.425% for language context, 0.471% for country and year of the movie release, 0.499% for director of the movie and 0.579% for the movie genre.
2) ***Multiplicative Model***: When kernels are non-linear and are multiplied point wise, then the results of including various contexts for item-based version ($KCR_{ub\otimes}$) are shown in Figure 7. This graph shows the decrease in RMSE for a user-based version of context aware algorithm when we multiply various contexts. The percentage decrease in RMSE for the contexts described in the graph is 0.017% for the director, 0.116% for language, country and year of release respectively.

Figure 7 shows the decrease in RMSE for a user-based version of context aware algorithm. The percentage decrease

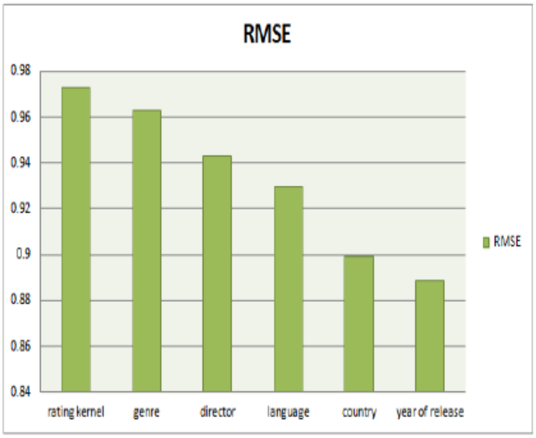

FIGURE 7. Results of different item context kernels in multiplicative KCR. X-axis = different context kernels, Y- axis = RMSE, dataset = LDOS-CoMoDa (full dataset) model = multiplicative KCR, version = user-based.

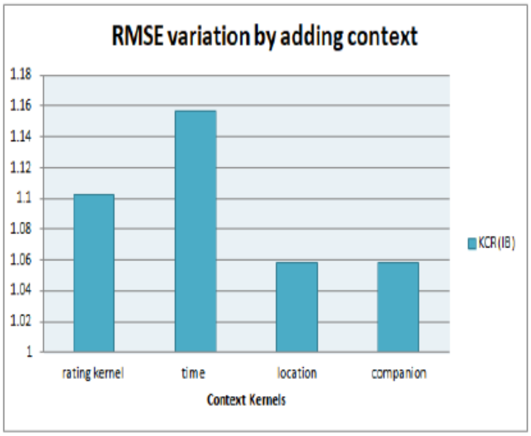

FIGURE 8. Results of various user context kernels in additive KCR in terms of RMSE over DePaulMovie dataset.

in RMSE for the contexts is 0.017% for director, 0.116% for the language, country and year of release respectively.

Results for the user based version explain the increase in performance of a system by representing the gradual decrease in RMSE.

B. EVALUATION PERFORMANCE WHEN DATASET IS DENSER

This scenario means that we are dealing with only those records, which have all the contextual values and hence the dataset is less sparser.

1) ITEM-BASED VERSION

This section describes the results of DepaulMovie dataset when it becomes denser. The context used here is user related i.e. time, location, companion.

1) *Additive Model* Figure 8 explain the effect of context for DepaulMovie dataset when kernels are linearly added.

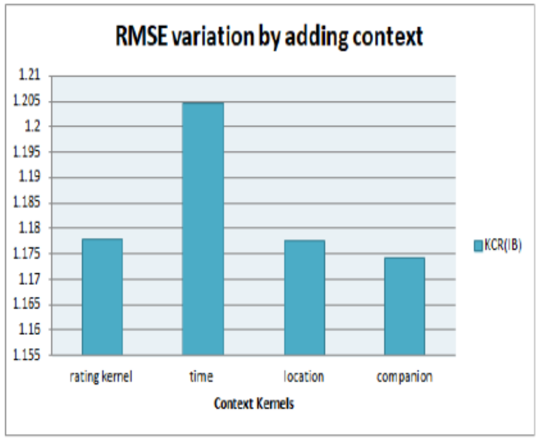

FIGURE 9. Results of various user context kernels in multiplicative KCR in terms of RMSE over DePaulMovie dataset.

The percentage decrease in RMSE in Figure 8 for various contexts described in the graph is 0.042% for location and 0.0449% for a companion.

2) *Multiplicative Model* When kernels are non-linearly multiplied then the results of the item based multiplied model for DepaulMovie dataset is described in Figure 9.

Figure 9 shows the gradual increase in performance by adding various contexts as 0.004% by adding location and 0.0035% by adding a companion feature vector.

From the above given results we observe that both additive and multiplicative models of our item-based version has good results i.e. when kernels are linear or non-linear.

C. COMPARISON OF KCR WITH KMR MODELS

In this section, we discuss the variation in RMSE by adding various contexts and then compare these results with the KMR models. This comparison reveals that the context is an important factor for providing accurate predictions. In the case of LDOS-CoMoDa dataset, we have compared results for both item- and user-based versions as we have both types of contexts i.e. user related and item related. However, in case of DePaulMovie dataset, we have only user-related context, so we are discussing only item-based version.

1) ITEM-BASED ADDITIVE MODEL

The graph given in Figure 10 empirically compares the performance of simple KMR algorithm with our proposed context-aware KMR framework in terms of RMSE for LDOS-CoMODa dataset. The gradual decrease in RMSE by adding context explains the importance of various contexts in the recommendation process.

2) ITEM-BASED MULTIPLICATIVE MODEL

Figure 11 describes the results of the various item-based multiplicative models for LDOS-CoMODa dataset in which all kernels are multiplied point-wise.

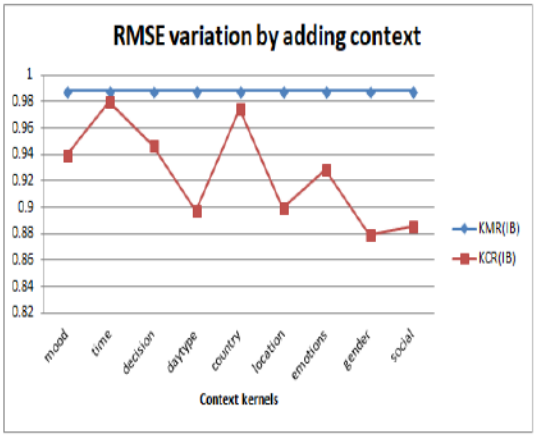

FIGURE 10. A comparison of item-based additive models in terms of root mean absolute error over LDOS-CoMODa dataset.

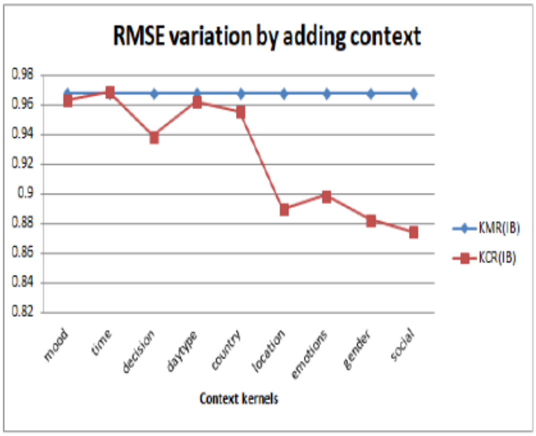

FIGURE 11. A comparison of item-based multiplicative models in terms of mean absolute error over LDOS-CoMODa dataset.

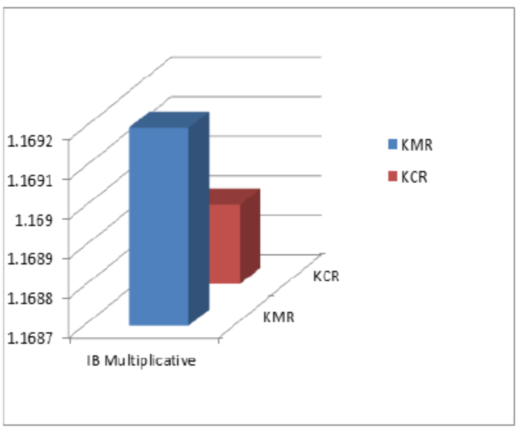

FIGURE 12. A comparison of item-based multiplicative models of KMR and KCR in terms of mean absolute error over DePaulMovie dataset.

Similarly, Figure 12 represents the results of DePaulMovie dataset in case of a multiplicative model of item based version.

3) USER-BASED ADDITIVE MODELS

Figure 13 compares the performance of additive and multiplicative models of simple KMR algorithm with our proposed context aware KMR framework in terms of RMSE. The gradual decrease in RMSE by adding context explains the importance of various contexts in the recommendation process. Figure 13 illustrates the results for LDOS-CoMODa dataset.

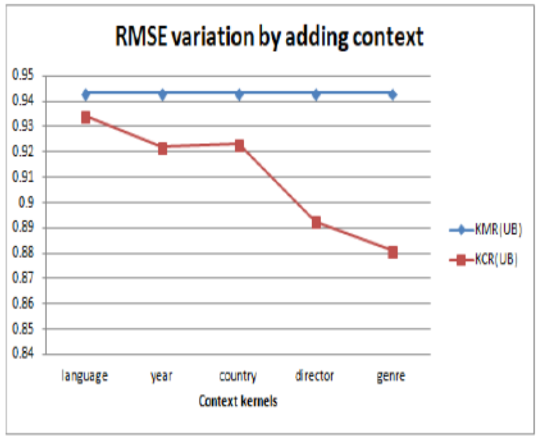

FIGURE 13. A comparison of user-based additive models in terms of root mean square error over LDOS-CoMODa dataset.

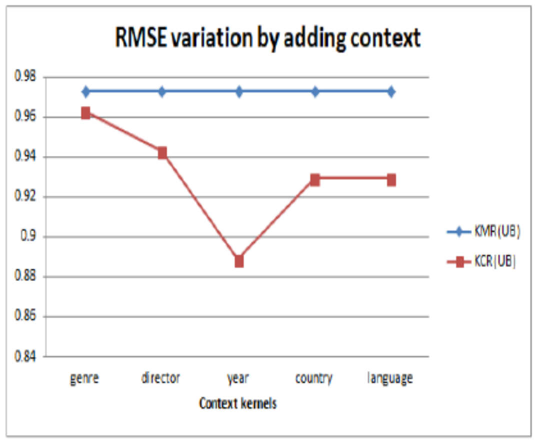

FIGURE 14. A comparison of user-based multiplicative models in terms of root mean square error over LDOS-CoMODa dataset.

4) USER-BASED MULTIPLICATIVE MODELS

Figure 14 describes the results of user based multiplicative models for both KMR and KCR models. Variation in root mean square error (RMSE) reveals which of the contextual information is more important than the others. Figure 14 is the demonstration of a multiplicative model of user-based version for LDOS-CoMODa dataset.

D. COMPARISON OF PROPOSED APPROACHES WITH OTHERS

We have compared our results with several other approaches such as Simple KMR (that is based on just rating kernel) and post-filtered context aware KMR, which is based on adding contextual kernels after training the model separately on these kernels. Post-filtering techniques [50] utilize context information only to adjust predictions, which are generated by a context-free 2D prediction model. The recommendation list is generated first and is further scanned for re-ranking and predictions based on certain context.

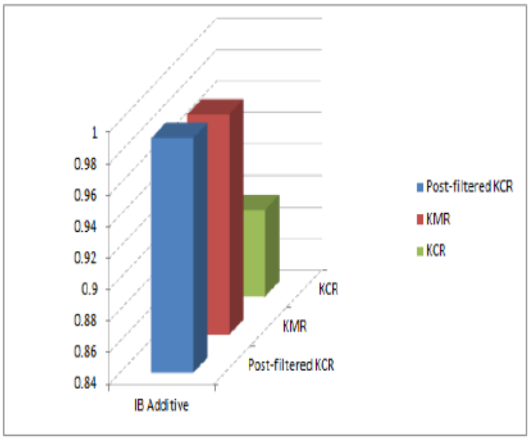

FIGURE 15. A comparison of Item based version of additive model of simple KMR, basic KCR framework and post-filtered KCR in terms of RMSE.

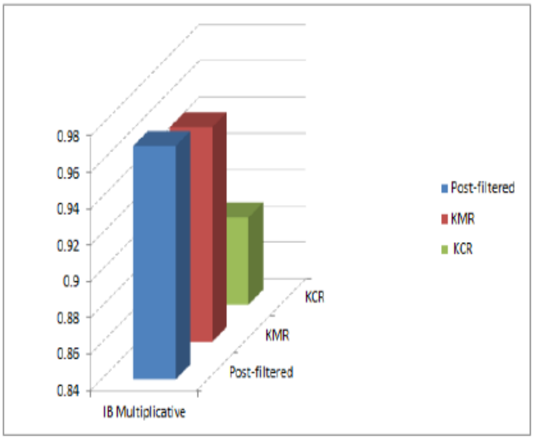

FIGURE 16. A comparison of item based version of multiplicative model of simple KMR, basic KCR framework and post-filtered KCR in terms of RMSE.

Figure 15, 16, 17 and in 18 demonstrate the improvement in performance by our proposed model. The increase in performance in item-based version is 0.091% for simple KMR and 0.146% for a post-filtering technique, which is a preferred approach in literature. The percentage increase in additive models of the simple user based KMR is 0.271% and 1.778% in post-filtering approach. The increase in accuracy in a multiplicative model of simple KMR model is 0.058% and 2.134% increase in the post-filtered contextual model. Table 5 provides a comparison of different evaluation parameters of our proposed model with the simple KMR model that is based on the concept of existing recommender systems, which only use ratings for the recommendation process. The best results are shown in bold font.

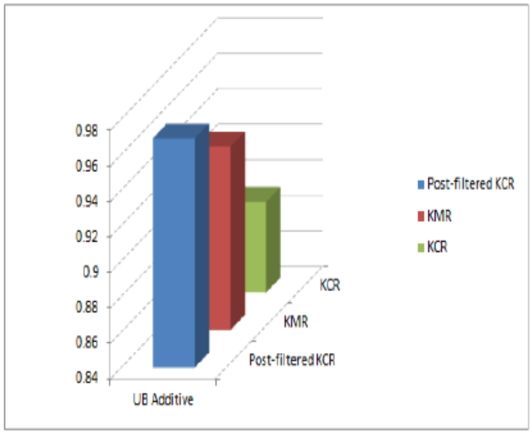

FIGURE 17. A comparison of user based version of additive model of simple KMR, basic KCR framework and post-filtered KCR in terms of RMSE.

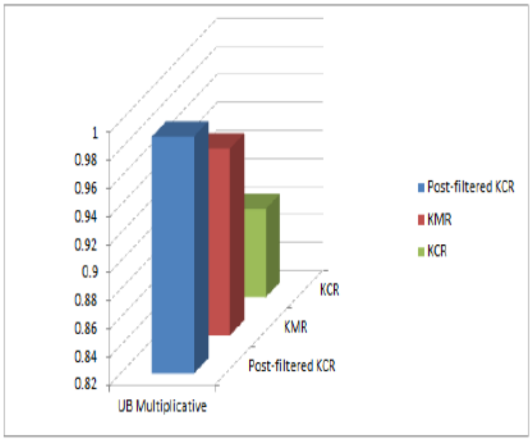

FIGURE 18. A comparison of user-based version of multiplicative model of simple KMR, basic KCR framework and KMR with post-filtered context in terms of RMSE.

TABLE 5. Results comparison of proposed context aware models for various evaluation parameters.

Result comparison in terms of evaluation parameters		
Approaches	RMSE	F1-Measure
$KMR_{ib\oplus}$	0.9880	0.1788
$KCR_{ib\oplus}$	0.8950	0.1807
$KMR_{ub\oplus}$	0.9433	0.2024
$KCR_{ub\oplus}$	0.8911	0.2023
$KMR_{ib\otimes}$	0.9682	0.1935
$KCR_{ib\otimes}$	0.8880	0.1927
$KMR_{ub\otimes}$	0.9731	0.2085
$KCR_{ub\otimes}$	**0.8832**	**0.2085**

A complete comparison of performance results over the whole dataset is described in Table 6.

Here we have compared our results with the other approaches proposed in the field of context-aware recommendations. Here we considered Context Aware Matrix

TABLE 6. A performance comparison of 3 different proposed approaches in terms of root mean square error (RMSE).

LDOS-CoMoDa dataset				
		Simple KMR	Post-filtered Model	Proposed Model
ADDITIVE MODELS	Item-Based Version	0.9880	0.9891	0.8950
	User-Based Version	0.9433	0.9694	0.8911
MULTIPLICATIVE MODELS	Item-Based Version	0.9682	0.9685	0.8880
	User-Based Version	0.9731	0.9896	**0.8832**

TABLE 7. A performance comparison with different approaches in terms of root mean square error (RMSE).

LDOS-CoMoDa dataset		
	Algorithms	RMSE
CAMF [44]	CAMF_C	1.0120
	$\text{CAMF}_C I$	1.0320
	$\text{CAMF}_C U$	0.9320
DCM [21]	DCR	1.0430
	DCW	1.0170
UI Splitting [20]	UBCF	1.0120
	IBCF	0.9721
	BiasCF	0.9270
TF and Context	Bias TF-RT [45]	0.9441
KCR	$KCR_{IB\oplus}$	0.8950
	$KCR_{UB\oplus}$	0.8911
	$KCR_{IB\otimes}$	0.8880
	$KCR_{UB\otimes}$	**0.8832**

Factorization Algorithm (CAMF) [44], Differential Context Modeling (DCM) Approach [21], Splitting Context aware approaches in [20] and proposed Bias TF-RT in [45] to show the comparison of results in terms of RMSE. Table 7 compares the results of our approach with others in terms of RMSE using the same dataset (LDOS-CoMoDa).

The best results of our proposed framework are shown in bold font. All such aforementioned results are calculated under various conditions when context kernels are built separately.

E. INCORPORATING VARIOUS CONTEXTUAL VECTORS AS A SINGLE CONTEXT KERNEL

There is another way to incorporate various contexts into rating kernel, where, we can concatenate various contextual vectors into a single vector and then build context kernel from that vector. Item-based and user-based, additive and multiplicative models are also proposed for this approach.

1) RESULT COMPARISON ON THE WHOLE DATASET

Results discussed below are evaluated on the whole dataset with users having all contextual values as well as those who have less or (some) missing contexts. This approach has better results for user-based version and is presented in Table 8.

The results presented in Figure 19, 20, 21, 22 show that the accuracy of various approaches increases. The proposed models have better results in various cases. Different results are mentioned in Table 9 and in Figure 23 and 24, in the case where all the kernels are changed to the polynomial kernel.

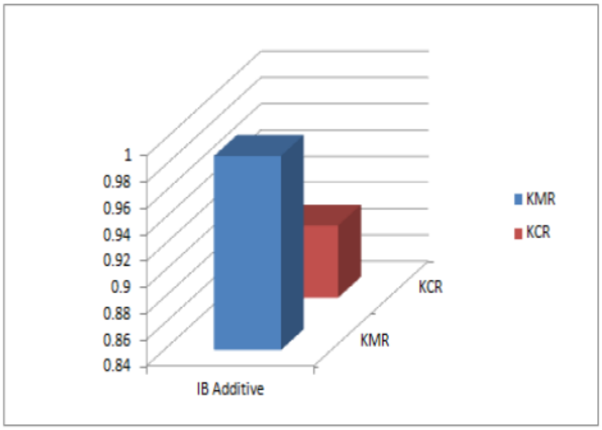

FIGURE 19. Decrease in RMSE for various item based approaches, dataset = LDOS-CoMoDa.

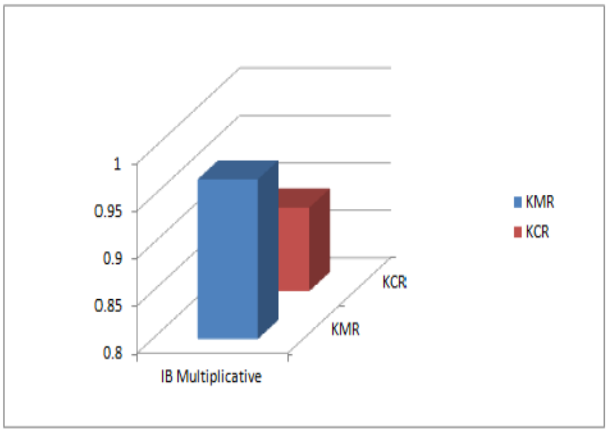

FIGURE 20. Decrease in RMSE for Item-based Multiplicative Approach, Dataset D LDOS-CoMoDa.

2) RESULT COMPARISON ON A DENSE DATASET

Empirically results of different models over dense dataset used in this work are discussed in this section. In other words, we can say that all the missing values are removed from the dataset and then various models were trained over this dataset. In such scenario, almost 1.09 % of the whole data in LDOS-CoMoDa dataset contains missing values, which are eliminated and make the data denser. Similarly, in the case of DePaulMovie dataset 75.2% of the whole data have missing values. So, by eliminating such type of records from the whole, we get much denser data. Comparison of the

TABLE 8. Result comparison when rating kernel is polynomial kernel and context kernel is Poly-Gaussian kernel.

LDOS-CoMoDa dataset			
		Simple KMR	Proposed Model
ADDITIVE MODELS	Item-Based Version	0.9880	0.8999
	User-Based Version	0.9433	0.8869
MULTIPLICATIVE MODELS	Item-Based Version	0.9682	**0.8781**
	User-Based Version	0.9731	0.8815

TABLE 9. Result comparison of user-based version when both rating kernel and context kernel is Polynomial kernel.

LDOS-CoMoDa dataset			
		Simple KMR	Proposed Model
ADDITIVE MODELS	User-Based Version	0.9402	0.8843
MULTIPLICATIVE MODELS	User-Based Version	0.9651	**0.8801**

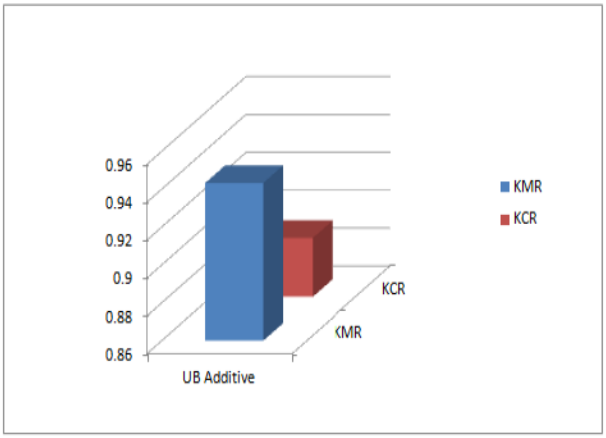

FIGURE 21. Decrease in RMSE for various user based additive model approaches, dataset = LDOS-CoMoDa.

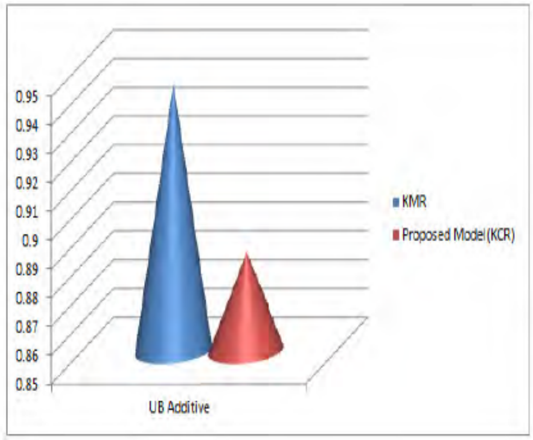

FIGURE 23. Decrease in RMSE of user based additive model by using proposed model, dataset = LDOS-CoMoDa.

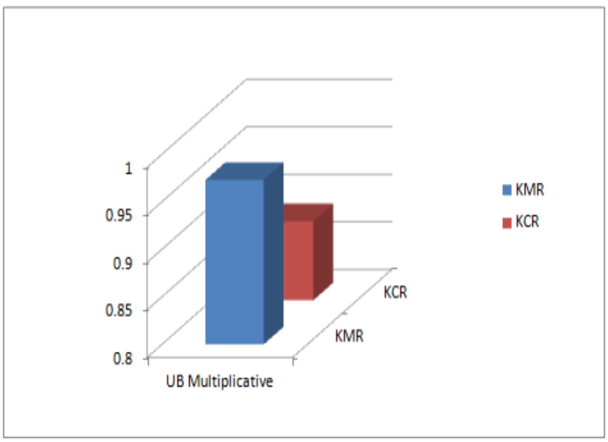

FIGURE 22. Decrease in RMSE for various user based multiplicative model approaches, dataset = LDOS-CoMoDa.

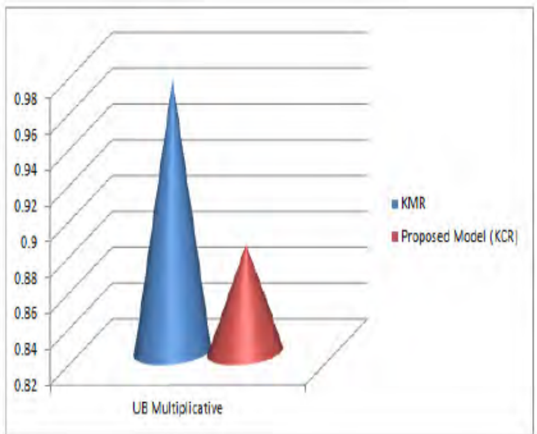

FIGURE 24. Decrease in RMSE of user based multiplicative model by using proposed model, dataset = LDOS-CoMoDa.

results are discussed in Table 10, 11 for LDOS-CoMoDa dataset. Figure 25 and Table 12, 13 illustrate the results for DePaulMovie dataset.

Tables 10, 11, 12, 13 demonstrate the performance of proposed contextual models under various conditions. When we use Polynomial kernel type for context kernels, then user-based versions for both additive and multiplicative models perform better as compared to item-based version. When context kernels are Poly-Gaussian kernel, then item-based additive model, user-based additive and multiplicative

TABLE 10. Comparing the results for rating and context kernels over LDOS-CoMoDa dataset. The Polynomial kernel has been used.

Result Comparison By Using Polynomial Kernel		
Approaches	RMSE	F1-Measure
$KMR_{ib\oplus}$	0.9811	0.1685
$KCR_{ib\oplus}$	0.8739	0.1701
$KMR_{ub\oplus}$	0.9402	0.1920
$KCR_{ub\oplus}$	0.8843	0.1999
$KMR_{ib\otimes}$	0.9745	0.1907
$KCR_{ib\otimes}$	**0.8712**	**0.1953**
$KMR_{ub\otimes}$	0.9651	0.1769
$KCR_{ub\otimes}$	0.8801	0.1795

TABLE 11. Comparing the results for rating and context kernels over LDOS-CoMoDa dataset. The Poly-Gaussian kernel has been used.

Result Comparison By Using Poly-Gaussian Kernel		
Approaches	RMSE	F1-Measure
$KMR_{ib\oplus}$	0.9811	0.1685
$KCR_{ib\oplus}$	0.8999	0.1721
$KMR_{ub\oplus}$	0.9402	0.1920
$KCR_{ub\oplus}$	0.8869	0.1866
$KMR_{ib\otimes}$	0.9795	0.1907
$KCR_{ib\otimes}$	**0.8781**	0.1895
$KMR_{ub\otimes}$	0.9640	0.1769
$KCR_{ub\otimes}$	0.8815	0.1856

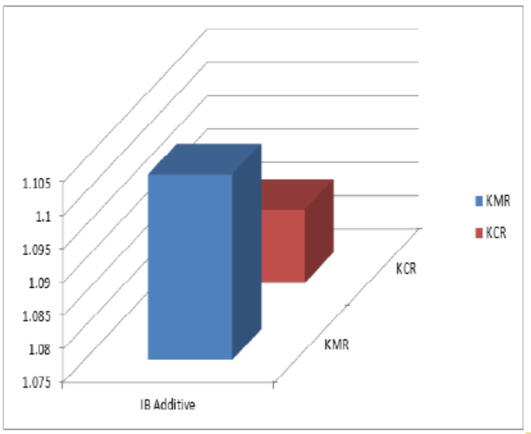

FIGURE 25. Decrease in Mean absolute error by adding contextual information, dataset = DePaulMovie.

TABLE 12. Comparing the results for rating and context kernels over DePaulMovie dataset. The polynomial kernel has been used.

Result Comparison by using polynomial kernel (Additive model)		
Approaches	RMSE 1	F1-Measure
$KMR_{ib\oplus}$	1.1028	0.3832
$KCR_{ib\oplus u^{context}_{time}}$	1.1564	0.3937
$KCR_{ib\oplus u^{context}_{loc}}$	1.0586	0.4222
$KCR_{ib\oplus u^{context}_{comp}}$	**1.0579**	**0.4349**
$KCR_{ib\oplus}$	1.0860	0.4180
$KMR_{ub\oplus}$	1.0582	0.5024
$KCR_{ub\oplus}$	1.0582	0.5024

models give better performance for various evaluation parameters. These results parameterized the importance of contexts like location, companions and time etc. in term of movie recommendations.

TABLE 13. Comparing the results for rating and context kernels over DePaulMovie dataset. The polynomial kernel has been used.

Result Comparison By Using Polynomial Kernel(Multiplicative Model)		
Approaches	RMSE	F1-Measure
$KMR_{ib\otimes}$	1.1779	0.3949
$KCR_{ib\otimes u^{context}_{time}}$	1.2045	0.3675
$KCR_{ib\otimes u^{context}_{loc}}$	1.1775	0.3949
$KCR_{ib\otimes u^{context}_{comp}}$	**1.1743**	**0.3949**
$KCR_{ib\otimes}$	1.2046	0.3705
$KMR_{ub\otimes}$	1.0976	0.4863
$KCR_{ub\otimes}$	1.0976	0.4863

VII. CONCLUSION AND FUTURE WORK

The research presented in this paper aims at proposing a novel context-aware framework, namely KCR, that is accurate, scalable and flexible enough to incorporate different contexts and can be used to make practical recommendations. Although the current state-of-the-art algorithms are quite accurate in their performance; however, these algorithms do not consider the context under which ratings are provided. A context is an important criterion in the domain of recommendations, so that the resultant predictions are closer to the user's taste. We have shown empirically, over two different datasets, that the proposed model performs better than the literature approaches (pre and post filtering) and moreover, provides good results on both the sparse and dense contextual dataset. Experiments demonstrate the claim of generality and flexibility of our approach, which enables the algorithm to incorporate contextual features—both user's and item's related— using kernel trick.

In future, the feature selection techniques like, principal component analysis (PCA) that can filter important context(s) from less important ones, might increase the performance of the proposed framework. Furthermore, we develop our algorithm by taking into account the cross-domain recommendations concepts, and we would like to extend our algorithm in this perspective, which ultimately would be a good increment in our framework. One potential future avenue is to enhance our algorithm by using the concept of pairwise kernels and deep learning. In this way, our algorithm will be able learn from different representations of data for various contexts, which might further improve its performance.

REFERENCES

[1] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, pp. 1–35.

[2] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of net-news," in *Proc. ACM Conf. Comput. Supported Cooperat. Work*, 1994, pp. 175–186.

[3] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens: Applying collaborative filtering to usenet news," *Commun. ACM*, vol. 40, pp. 77–87, Mar. 1997.

[4] J. Raymond Mooney and L. Roy, "Content-based book recommending using learning for text categorization," in *Proc. 5th ACM Conf. Digit. Libraries (DL)*, San Antonio, TX, USA, 2000, pp. 195–204.

[5] M. Mandl, A. Felfernig, E. Teppan, and M. Schubert, "Consumer decision making in knowledge-based recommendation," *J. Intell. Inf. Syst.*, vol. 37, no. 1, pp. 1–22, 2011.

[6] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure, "Ontological user profiling in recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 54–88, Jan. 2004.
[7] Y. Wang, S. C.-F. Chan, and G. Ngai, "Applicability of demographic recommender system to tourist attractions: A case study on trip advisor," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol. (WI-IAT)*, Washington, DC, USA, vol. 3, Dec. 2012, pp. 97–101.
[8] R. Burke, "Hybrid Web recommender systems," in *The Adaptive Web*. Berlin, Germany: Springer-Verlag, 2007, pp. 377–408.
[9] M. A. Ghazanfar, "Experimenting switching hybrid recommender systems," *Intell. Data Anal.*, vol. 19, no. 4, pp. 845–877, 2015.
[10] J. P. Lucas, N. Luz, M. N. Moreno, R. Anacleto, A. A. Figueiredo, and C. Martins, "A hybrid recommendation approach for a tourism system," *Expert Syst. Appl.*, vol. 40, no. 9, pp. 3532–3550, 2013.
[11] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.
[12] M. A. Ghazanfar and A. Prügel-Bennett. "Exploiting context in kernel-mapping recommender system algorithms," in *Proc. 6th Int. Conf. Mach. Vis. (ICMV)*, 2013, p. 906727.
[13] M. A. Ghazanfar and A. Prügel-Bennett. "Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3261–3275, 2014.
[14] M. A. Ghazanfar, A. Prügel-Bennett, and S. Szedmak. "Kernel-mapping recommender system algorithms," *Inf. Sci.*, vol. 208, pp. 81–104, Nov. 2012.
[15] X. Yu, Y. Chu, F. Jiang, Y. Guo, and D. Gong, "A user-based cross domain collaborative filtering algorithm based on a linear decomposition model," *IEEE Access*, vol. 5, pp. 27582–27589, 2017.
[16] X. Yu, Y. Chu, F. Jiang, Y. Guo, and D. Gong, "SVMs classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features," *Knowl.-Based Syst.*, vol. 141, pp. 80–91, Feb. 2018.
[17] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2011, pp. 635–644.
[18] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, Jul. 2013.
[19] K. Verbert *et al.*, "Context-aware recommender systems for learning: A survey and future challenges," *IEEE Trans. Learn. Technol.*, vol. 5, no. 4, pp. 318–335, Oct./Dec. 2012.
[20] Y. Zheng, R. Burke, and B. Mobasher, "The role of emotions in context-aware recommendation," in *Proc. ACM 3rd Int. Workshop Hum. Decis. Making Recommender Syst. (RecSys)*, 2013, pp. 21–28.
[21] Y. Zheng, R. Burke, and B. Mobasher, "Recommendation with differential context weighting," in *User Modeling, Adaptation, and Personalization* (Lecture Notes in Computer Science), vol. 7899. Berlin, Germany: Springer, 2013, pp. 152–164.
[22] L. Baltrunas, "Context-aware collaborative filtering recommender systems," Ph.D. dissertation, Dept. Comput. Sci., Free Univ. Bozen-Bolzano, Bolzano, Italy, 2011.
[23] A. Schmidt, M. Beigl, and H.-W. Gellersen, "There is more to context than location," *Comput. Graph.*, vol. 23, pp. 893–901, 1999.
[24] C. Hayes and P. Cunningham, "Context boosting collaborative recommendations," *Knowl.-Based Syst.*, vol. 17, no. 2, pp. 131–138, 2004.
[25] L. Baltrunas, "Exploiting contextual information in recommender systems," in *Proc. ACM Conf. Recommender Syst.*, 2008, pp. 295–298.
[26] L. Baltrunas and X. Amatriain, "Towards time-dependant recommendation based on implicit feedback," in *Proc. Workshop Context-Aware Recommender Syst. (CARS)*, 2009, pp. 1–5.
[27] P. G. Campos, F. Díez, and I. Cantador. "Time-aware recommender systems: A comprehensive survey and analysis of existing evaluation protocols," *User Model. User-Adapted Interact.*, vol. 24, nos. 1–2, pp. 67–119, 2014.
[28] T. Cebrián, M. Planagumà, P. Villegas, and X. Amatriain, "Music recommendations with temporal context awareness," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 349–352.
[29] Z. Gantner, S. Rendle, and L. Schmidt-Thieme, "Factorization models for context-/time-aware movie recommendations," in *Proc. Workshop Context-Aware Movie Recommendation*, 2010, pp. 14–19.
[30] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 79–86.
[31] A. Beach *et al.*, "Fusing mobile, sensor, and social data to fully enable context-aware computing," in *Proc. 11th Workshop Mobile Comput. Syst. Appl.*, 2010, pp. 60–65.
[32] D. Lee, S. E. Park, M. Kahng, S. Lee, and S.-G. Lee, "Exploiting contextual information from event logs for personalized recommendation," in *Computer and Information Science*. Berlin, Germany: Springer, 2010, pp. 121–139.
[33] H.-S. Park, J.-O. Yoo, and S.-B. Cho, "A context-aware music recommendation system using fuzzy bayesian networks with utility theory," in *Fuzzy Systems and Knowledge Discovery*. Berlin, Germany: Springer, 2006, pp. 970–979.
[34] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Trans. Inf. Syst.*, vol. 23, no. 1, pp. 103–145, 2005.
[35] S. S. Anand and B. Mobasher, *Contextual Recommendation*. Berlin, Germany: Springer-Verlag, 2007.
[36] S. Berkovsky *et al.*, "Predicting user experiences through cross-context reasoning," in *Proc. LWA*, 2006, pp. 27–31.
[37] Y. Zheng, R. Burke, and B. Mobasher, "Context-aware splitting approaches: Split users or split items?" in *Research Colloquium*. Chicago, IL, USA: DePaul Univ., 2013.
[38] Y. Zheng, B. Mobasher, and R. Burke, "Emotions in context-aware recommender systems," in *Emotions and Personality in Personalized Services*. Berlin, Germany: Springer, 2016, pp. 311–326.
[39] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan. 2003.
[40] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 426–434.
[41] N. Srebro, J. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1329–1336.
[42] S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 81–90.
[43] M. A. Ghazanfar and A. Prügel-Bennett, "Exploiting context in kernel-mapping recommender system algorithms," in *Proc. 6th Int. Conf. Mach. Vis. (ICMV)*, 2013, p. 906727.
[44] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proc. 5th ACM Conf. Recommender Syst.*, 2011, pp. 301–304.
[45] W. Wu *et al.*, "Improving performance of tensor-based context-aware recommenders using Bias Tensor Factorization with context feature auto-encoding," *Knowl.-Based Syst.*, vol. 128, pp. 71–77, Jul. 2017.
[46] M. Gorgoglione, C. Palmisano, U. Panniello, P. Antonio, and A. Alexander, "Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems," in *Proc. 3rd ACM Conf. Recommender Syst.*, 2009, pp. 265–268.
[47] T. Joachims, "Training linear SVMs in linear time," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 217–226.
[48] S. Szedmak, Y. Ni, and S. R. Gunn, "Maximum margin learning with incomplete data: Learning networks instead of tables," in *Proc. WAPA*, 2010, pp. 96–102.
[49] L. Cui, W. Huang, Q. Yan, F. R. Yu, Z. Wen, and N. Lu, "A novel context-aware recommendation algorithm with two-level SVD in social networks," *Future Gener. Comput. Syst.*, vol. 86, pp. 1459–1470, Sep. 2018.
[50] H. Wu, K. Yue, X. Liu, Y. Pei, and B. Li, "Context-aware recommendation via graph-based contextual modeling and postfiltering," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 8, 2015, Art. no. 613612.

MISBAH IQBAL received the M.Sc. degree in software engineering from the University of Engineering and Technology, Taxila, Pakistan, in 2015, and the M.Sc. degree in computer networking from the University of Bedfordshire, Luton, U.K. She is currently a student of International MBA with BPP University, London, U.K. Her areas of interests include machine learning, artificial intelligence, computer networking, human resource management, and business operations and management.

MUSTANSAR ALI GHAZANFAR received the Ph.D. degree in artificial intelligence from the University of Southampton, U.K. He has over 10 years of experience in industry, and research and development. He has published more than 40 international articles. His areas of interests include recommender systems, predicting modeling, and applied machine/deep learning.

ASMA SATTAR received the master's degree in software engineering from the University of Engineering and Technology, Taxila, in 2015. She is currently an Assistant Professor with the National University of Computer and Emerging Sciences, Chiniot-Faisalabad Campus, Pakistan. Her research interests include mobile application development, machine learning, and recommender systems.

MUAZZAM MAQSOOD received the Ph.D. degree from the University of Engineering and Technology, Taxila, in 2017. He is currently an Assistant Professor with COMSATS University Islamabad, Attock Campus, Pakistan. His research interests include machine learning, recommender systems, and image processing.

SALABAT KHAN received the Ph.D. degree from FAST, Islamabad. He is currently an Assistant Professor with COMSATS University Islamabad, Attock Campus, Pakistan. His research interests include machine learning, optimization, and data mining.

IRFAN MEHMOOD has been involved in IT industry and academia in Pakistan and South Korea for a decade. He is currently an Assistant Professor with the Software Department, Sejong University. His sustained contribution at various research and industry-collaborative projects gives him an extra edge to meet the current challenges faced in the field of multimedia analytics. Specifically, he has made significant contribution in the areas of video summarization, medical image analysis, visual surveillance, information mining, deep learning in industrial applications, and data encryption.

SUNG WOOK BAIK (M'16) received the B.S. degree in computer science from Seoul National University, Seoul, South Korea, in 1987, the M.S. degree in computer science from Northern Illinois University, Dekalb, IL, USA, in 1992, and the Ph.D. degree in information technology engineering from George Mason University, Fairfax, VA, USA, in 1999. He was with Datamat Systems Research Inc., as a Senior Scientist of the Intelligent Systems Group, from 1997 to 2002. In 2002, he joined the faculty of the College of Electronics and Information Engineering, Sejong University, Seoul, where he is currently a Full Professor and the Dean of Digital Contents. He is also the Head of the Intelligent Media Laboratory, Sejong University. His research interests include computer vision, multimedia, pattern recognition, machine learning, data mining, virtual reality, and computer games.