MobiContext: A Context-aware Cloud-Based Venue Recommendation Framework

Rizwana Irfan, Osman Khalid, Muhammad Usman Shahid Khan,
Camelia Chira, Rajiv Ranjan, Fan Zhang, *Senior Member, IEEE,*
Samee U. Khan, *Senior Member, IEEE,* Bharadwaj Veeravalli, *Senior Member, IEEE,*
Keqin Li, *Fellow, IEEE,* and Albert Y. Zomaya, *Fellow, IEEE*

Abstract— In recent years, recommendation systems have seen significant evolution in the field of knowledge engineering. Most of the existing recommendation systems based their models on collaborative filtering approaches that make them simple to implement. However, performance of most of the existing collaborative filtering-based recommendation system suffers due to the challenges, such as: (a) cold start, (b) data sparseness, and (c) scalability. Moreover, recommendation problem is often characterized by the presence of many conflicting objectives or decision variables, such as users' preferences and venue closeness. In this paper, we proposed *MobiContext*, a hybrid cloud-based Bi-Objective Recommendation Framework (BORF) for mobile social networks. The *MobiContext* utilizes multi-objective optimization techniques to generate personalized recommendations. To address the issues pertaining to cold start and data sparseness, the BORF performs data preprocessing by using the Hub-Average (HA) inference model. Moreover, the Weighted Sum Approach (WSA) is implemented for scalar optimization and an evolutionary algorithm (NSGA-II) is applied for vector optimization to provide optimal suggestions to the users about a venue. The results of comprehensive experiments on a large-scale real dataset confirm the accuracy of the proposed recommendation framework.

Index Terms— Multi-objective optimization, Collaborative Filtering (CF), Non-dominated Sorting Genetic Algorithm (NSGA-II).

1 INTRODUCTION

THE ongoing rapid expansion of the Internet and easy availability of numerous e-commerce and social networks services, such as *Amazon*, *Foursquare*, and *Gowalla*, have resulted in the sheer volume of data collected by the service providers on daily basis. The continuous accumulation of massive volumes of data has shifted the focus of research community from the basic information retrieval problem to the filtering of pertinent information [1], thereby making it more relevant and personalized to user's query. Therefore, most research is now directed towards the designing of more intelligent and autonomous information retrieval systems, known as *Recommendation Systems.*

- *R. Irfan, M. U. S. Khan, and S. U. Khan are with the North Dakota State University, Fargo, ND 58108, USA. Email:{rizwana.irfan, ushahid.khan, samee.khan}@ndsu.edu.*
- *O. Khalid is with the COMSATS Institute of Information Technology, Pakistan. Email:{osman@ciit.net.pk}*
- *C. Chira is with the Technical University Cluj-Napoca, Romania. E-mail: cchira@cs.ubbcluj.ro.*
- *R. Ranjan is with ICT Centre, CSIRO, Canberra, Australia. E-mail: raj.ranjan@csiro.au.*
- *F. Zhang is with the Massachusetts Institute of Technology, MA 02139, USA. E-mail: f_zhang@mit.edu.*
- *B. Veeravalli is with the National University of Singapore. E-mail: elebv@nus.edu.sg.*
- *K. Li is with the State University of New York at Paltz. E-mail: lik@newpaltz.edu.*
- *A. Zomaya is with the University of Sydney. E-mail: albert.zomaya@sydney.edu.au.*

1.1 Research Motivation

Recommendation systems are increasingly emerging as an integral component of e-business applications [1]. For instance, the integrated recommendation system of Amazon provides customers with personalized recommendations for various items of interest. Recommendation systems utilize various knowledge discovery techniques on a user's historical data and current context to recommend products and services that best match the user's preferences.

In recent years, emergence of numerous mobile social networking services, such as, *Facebook* and *Google Latitude* has significantly gained the attraction of a large number of subscribers [1], [6]. A mobile social networking service allows a user to perform a "check-in" that is a small feedback about the place visited by the user [1], [2], [22]. Large number of check-ins on daily bases results in the accumulation of massive volumes of data. Based on the data stored by such services, several Venue-based Recommendation Systems (VRS) were developed [1]–[3]. Such systems are designed to perform recommendation of venues to users that most closely match with users' preferences. Despite having very promising features, the VRS suffer with numerous limitations and challenges. A major research challenge for such systems is to process data at the real-time and extract preferred venues from a massively huge and diverse dataset of users' historical check-ins [1]–[3], [12], [13]. Further complexity to the problem is added by also taking into the account the real-time contextual information, such as: (a) venue selection based on user's personal preferences and (b) venue closeness based on geographic information.

1.2 Research Problem

In scientific literature, several works, such as [1]–[6], and [13] have applied Collaborative Filtering (CF) to the recommendation problem in VRS. The CF-based approaches in VRS tend to generate recommendations based on the similarity in actions and routines of users [1], [2], [5]. However, despite being less complicated, most CF-based recommendation techniques suffer from several limitations that make them less ideal choice in many real-life practical applications [13]. The following are the most common factors that affect the performance of many existing CF-based recommendation systems:

- *Cold start*. The cold start problem occurs when a recommendation system has to suggest venues to the user that is newer to the system [2]. Insufficient check-ins for the new user results in zero similarity value that degrades the performance of the recommendation system [13]. The only way for the system to provide recommendation in such scenario is to wait for sufficient check-ins by the user at different venues.
- *Data sparseness.* Many existing recommendation systems suffer from data sparseness problem that occurs when users have visited only a limited number of venues [3]. This results into a sparsely filled user-to-venu check-in matrix. The sparseness of such matrix creates difficulty in finding sufficient reliable similar users to generate good quality recommendation.
- *Scalability.* Majority of traditional recommendation systems suffer from scalability issues. The fast and dynamic expansion of number of users causes recommender system to parse millions of check-in records to find the set of similar users. Some of the recommendation systems [2], [3], [24] employ data mining and machine learning techniques to reduce the dataset size. However, there is an inherent tradeoff between reduced dataset size and recommendation quality [1].

The immediate effect of the above-mentioned issues is the degradation in performance of most of the CF-based recommendation systems. Therefore, it is not adequate to rely solely on simplistic but memory-intensive CF approach to generate recommendations.

1.3 Methods and Contributions

In this paper, we propose *MobiContext*, a hybrid cloud-based Bi-Objective Recommendation Framework (BORF) that overcomes the limitations exhibited by traditional CF-based approaches. The *MobiContext* framework combines memory-based and model-based approach of CF in a hybrid architecture to generate optimal recommendations for the current user. The memory-based CF model utilizes a user's historical data and user-to-venue closeness to predict venues for the current user. To address data sparseness caused by zero similarities, we utilize a metric known as *confidence measure*. The confidence measure defines the conditional probability that two users will show interest in the same set of venues and is expressed as the ratio of the number of venues visited by both users together to the number of venues visited by any one of the two users [6]. The confidence measure is utilized to compute link weight between two users, if and only if the similarity between the users is zero. In this way, confidence measure helps replacing many zero similarity entries in user-to-user to matrix by alternate non-zero entries, thereby improving recommendation quality. The proposed framework also suggests a solution to cold start problem by utilizing model-based Hub-Average (HA) inference method [9]. The HA method computes and assigns popularity ranking to venues and users at various geographical locations. With such ranking available, the new user can be recommended with venues that have highest ranking in a geographical region.

To improve scalability performance, the cloud-based *MobiContext* framework follows Software as a Service (SaaS) approach by utilizing a modular service architecture. The primary advantage of this approach is that the proposed framework can scale on demand as additional virtual machines are created and deployed.

We adopt a bi-objective optimization approach that considers the two primary objectives: (a) venue preference and (b) location closeness. Venue preference determines how much the venue meets the criteria of user's interests, whereas venue closeness indicates how closely a desired venue is located relative to a user's location. The *MobiContext* framework generates optimized recommendations by simultaneously considering the trade-offs between the aforementioned objectives. In summary, the contributions of our work are as follows.

- We propose a cloud-based framework consisting of bi-objective optimization methods named as CF-BORF and greedy-BORF. The Genetic Algorithm based BORF (GA-BORF) utilizes Non-dominated Sorting Genetic Algorithm (NSGA-II) [16] to optimize the venue recommendation problem.
- We introduce a pre-processing phase that performs data refinement using HA.
- We perform extensive experiments on our internal OpenNebula cloud setup running on 96 core Supermicro SuperServer SYS-7047GR-TRF systems. The experiments were conducted on real-world "Gowalla" dataset [3].

To the best of our knowledge this is the first work to incorporate the bi-objective optimization techniques into VRS. The rest of the paper is organized as follows. Section 2 presents the system overview. In Section 3, we discuss the proposed BORF framework. Section 4 presents the complexity analysis of the proposed framework and the performance evaluation with simulation results. The related work is reviewed in Section 5, and Section 6 concludes the paper.

2 System Overview

Most of the existing recommendation systems (e.g., [2], [3], [5], [7]) utilize centralized architectures that are not scalable enough to process large volume of geographically distributed data. The centralized

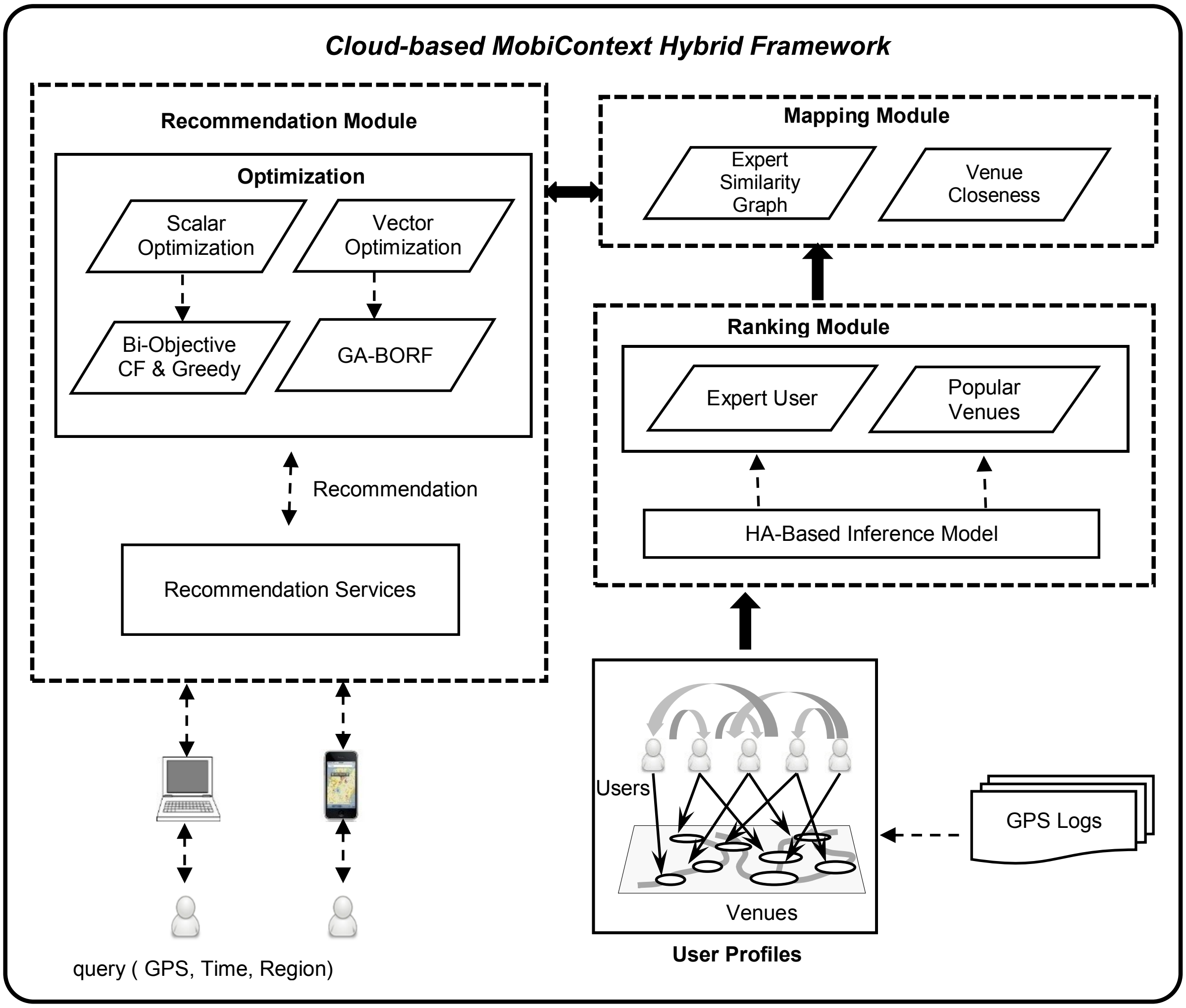


Fig. 1. Top level architecture of the Cloud-based MobiContext BORF framework

architecture for venue recommendations must simultaneously consider users' preferences, check-in history, and social context to generate optimal venue recommendations. Therefore, to address the scalability issue, we introduce the decentralized cloud-based *MobiContext* BORF approach. The following are some of the major components of the proposed framework.

2.1 User Profiles

As reflected in Fig. 1, the *MobiContext* framework maintains records of users' profiles for each geographical region. The arrows from users to venues at lower right of Fig. 1 indicate the number of check-ins performed by each user at various venues. A user's profile consists of the user's identification, venues visited by the user, and check-in time at a venue.

2.2 Ranking Module

On top of users' profiles, the ranking module performs functionality during the pre-processing phase of data refinement. The pre-processing can be performed in the form of periodic batch jobs running at monthly or weekly basis as configured by system administrator. The ranking module applies model-based HA inference method on users' profiles to assign ranking to the set of users and venues based on mutual reinforcement relationships [9]. The idea is to extract a set of popular venues and expert users. We call a venue as popular, if it is visited by many expert users, and a user as expert if (s)he has visited many popular venues [9], [15]. The users and venues that have very low scores are pruned from the dataset during offline pre-processing phase to reduce the online computation time.

2.3 Mapping Module

The mapping module computes similarity graphs among expert users for a given region during pre-processing phase. The purpose of similarity graph computation is to generate a network of like-minded people who share the similar preferences for various venues they visit in a geographical region. The mapping module also computes venue closeness based on geographical distance between the current user and popular venues.

2.4 Recommendation Module

Fig. 1 depicts the online recommendation module that runs a service to receive recommendation queries from users. A user's request consists of: (a) current context (such as, GPS location of user, time, and region), and (b) a bounded region surrounding the user from where the top

N venues will be selected for the current user (N is number of venues). The recommendation service passes the user's query to optimization module that utilizes scalar and vector optimization techniques [13], [15], [16] to generate an optimal set of venues. In our proposed framework, the scalar optimization technique utilizes the CF-based approach and greedy heuristics to generate user preferred recommendations. The vector optimization technique, namely GA-BORF, utilizes evolutionary algorithms, such as NSGA-II [16] to produce optimized recommendations. The detailed description of proposed optimization techniques is provided in Section 3.

3 MobiContext Recommendation Framework

In this section, we discuss in detail the functionality of the proposed *MobiContext* framework. The frequently used acronyms in this paper are listed in Table 1. In terms of functionality, *MobiContext* framework has two main phases: (a) a pre-processing phase and (b) a recommendation phase. The detailed description of the above mentioned phases is presented in the following subsequent sections.

3.1 Pre-processing Phase

Pre-processing is further divided into two phases: (a) ranking phase and (b) mapping phase, as described in the following subsections.

3.1.1. Ranking Module

The HA inference model is applied on users' profiles to compute ranking for users and venues. The higher ranked venues and users are known as popular venues and expert users, respectively. The framework maintains region-wise user-to-venue check-in matrix M_c (defined in Table 1) that is utilized to compute popularity ranking scores for users and venues. Let $[p_v]$ and $[e_u]$ represent score matrices for a popular venue and an expert user, respectively, for a region R. The following formulas compute the score for popular venues and expert users:

TABLE 1
NOTATIONS AND THEIR MEANINGS

Symbols	Meaning
$r_{c,v}.r_{c',v}$	Number of check-in at venue v performed by the user c and c'
M_c	Venue check-in matrix M for user c *(rows representing users and columns representing venues)*
V	Set of all venues
E	Set of expert user in a region
∂	Total number of popular venues checked-in by expert users
p_v	Popular venues
$\overline{r_c}$	Average number of check-ins of user c
$s_r(c,c')$	Similarity matrix of user c and c'
$s_{clo}(c,c'_v)$	Proximity matrix of user c and c′ with respect to venue v visited by user c′

$$p_v = M_c^T \times e_u. \tag{1}$$

$$e_u = M_c \times p_v \times \frac{1}{\partial}. \tag{2}$$

If we use $p_v^{<n>}$and $e_u^{<n>}$ to represent the scores of popular venues and expert users at nth iteration, then the following equations generate the score of popular venues and expert users iteratively.

$$p_v^{<n>} = (M_c^T \times M_c) \times p_v^{<n-1>}. \tag{3}$$

$$e_u^{<n>} = (M_c \times M_c^T) \times e_u^{<n-1>} \times \frac{1}{\partial}. \tag{4}$$

The purpose of using HA method is to generate a subset of users, who have visited popular venues, and a subset of venues that are frequently visited by expert users.

3.1.2. Mapping Module

The mapping phase computes the similarity among the expert users (that were generated by the ranking phase) using Pearson Correlation Coefficient (PCC). The PCC is widely used in recommendation systems to generate similarity graphs among users [13]. The graph constructed in the mapping phase will be made available for online recommendations.

The value of the PCC ranges between -1 and +1, where the value close to 1 indicates the higher degree of similarity exists between two users. If the value of PCC is zero or less than zero, then this means the preferences of two users (c and c') do not match. The PCC is computed by using the following formula.

$$s_r(c,c^{'}) = \frac{\sum_{v\epsilon S_{cc'}}(r_{cv}-\overline{r_c})(r_{c'v}-\bar{r}_{c'})}{\sqrt{\sum_{v\epsilon S_{cc'}}(r_{cv}-\overline{r_c})^2\sum_{v\epsilon S_{cc'}}(r_{c'v}-\bar{r}_{c'})^2}}, \tag{5}$$

where

$$S_{cc'} = \{v\epsilon V | r_{cv} \neq 0 \wedge r_{c'v} \neq 0\}.$$

In the above equation, the similarity between two experts c and c' is calculated only those venues that are visited by both the users. The similarity calculation in (5) results into a very sparse similarity graph because, majority of the venues are not visited by either of the two users. To address the data sparseness problem, we augment the similarity computation with the confidence measure. The confidence measure can be interpreted as a conditional probability that a venue visited by a one user is also visited by the other user in the dataset. The following equation is utilized to calculate the weight of an edge between two users.

$$\omega_{cc'} = \begin{cases} s_r(c,c') & if\ s_r(c,c') > 0 \\ & \text{otherwise} \\ P(r_c|r_{c'}) \times \dfrac{1}{1+\sum_{v\epsilon r_{c'}}|r_{cv}-r_{c'v}|}. & \end{cases} \tag{6}$$

Here, the parameters r_c and $r_{c'}$ are the set of venues checked-in by the user c and c', respectively. The parameter $P(r_c|r_{c'}) = P[r_c \cap r_{c'}]/P[r_{c'}]$ is the likelihood ratio that both the users may visit the similar set of venues in future. Moreover, the parameter $P[r_{c'}] \neq 0$ indicates the probability that the user c' performed check-in at $r_{c'}$ set of venues. The additional sum factor in denominator is used to keep value of probability lower than similarity so that the preference must be given to the positive values of similarity. Moreover, in (6), if the similarity value is greater than 0, then this value is assigned as an edge weight on the similarity graph. However, when the similarity value is less than zero, then we consider the lower term of (6) to assign the edge weight. This implies that an edge is always assigned a non-zero weight that helps reducing data sparseness due to zero similarities.

The mapping phase also computes the geographical distance of the current user from the popular venues [23], [24]. The geospatial information about current users and venues are presented as GPS coordinates. Therefore, we utilize Haversine model [8] to compute the user-to-venue distance as follows:

$$d = Ra, \tag{7}$$

where the parameter a is the angular distance in radians between the current user and the venue's geospatial location. The parameter R is earth's radius. We use a simple transformation function $s_{clo}(c, c'_v)$ to calculate the user-to-venue geographical closeness by taking inverse of the distance d, where c'_v is the venues visited by the user c'. The region-wise similarity graph of expert users and location closeness of popular venues are stored in the database for later online recommendation phase.

3.2 Recommendation Module

The online recommendation module utilizes bi-objective optimization to generate an optimized list of venues. Suppose an current user A is interested in venue type T that must be located closest to the current location of the current user within a specific region R. In such a scenario, the current user requires the best preferred venues as well as the closest venues from the user's current location. To meet both the aforementioned objectives, we utilize bi-objective optimization in the proposed *MobiContext* recommendation framework. The optimization module simultaneously maximizes the following two objectives: (a) popular venues and (b) venues' closeness that can be stated as:

$$max\ f(o_i)\ \forall\ o_i \in \{\, p_v, v_c \,\}, \tag{8}$$

where, the parameter $f(o_i)$ represents the maximized objective function, in terms of popular venues visited by expert users (p_v) and venue closeness (v_c). In the subsequent subsection, we discuss the approaches we utilized to address the bi-objective optimization in the *MobiContext* framework.

Algorithm 1. CF- BORF-based Venue Selection

Input: Current User: c, region: R

Output: Top_{rec}= A set S' of top-N venues.

Definitions, V_e= *set of venues visited by expert user e*, N_c= *set of recommended venues*, l_c=*location of current user c*, V_c = *set of venues visited by current user*, S_r = *set of expert users similar to the current user c*, ς_{ce} = *closeness measure of the expert user e with the location of current user c*, s_{ce} *is similarity of the user c with the expert user e.*

1: $N_c \leftarrow \emptyset$; $z_{agg} \leftarrow \emptyset$;
2: $S_r \leftarrow computsimset$ (c, E)
3: **for** *each* $e \in S_r$ do
4: S ← $\{v : V_e | v \notin V_c\}$
5: $\varsigma_{ce} \leftarrow max(computsimD(l_c, \mathrm{S}))$
6: $z_{agg}[\mathrm{e}] \leftarrow computeagg(s_{ce}, \varsigma_{ce})$
7: **end for**
8: $N_c \leftarrow computRec(\mathrm{c},\ z_{agg})$
9: $Top_{rec} \leftarrow sort\ (N_c)$

3.2.1 Scalar Optimization

Our first recommendation approach is based on state-of the-art scalar optimization technique [17] that transforms the multiple objectives into a single-objective aggregate function. For such transformation, we utilized weighted-sum approach because of its simplicity, ease of use, and direct translation of weight into the relative importance of the objectives [18]. The weighted sum approach for BORF can be presented as follows.

$$f(u) = \sum_{i=1}^{n} \alpha_i \times f_i(u), \tag{9}$$

where the function $f(u)$ is the aggregate objective function, the parameter α_i is the weight that determines the significance of n number of objective functions [11], [16]. In our scenario, there are two objective functions, termed as preferred venue and venue closeness. The weights for preferred venue and venue closeness are formulated in the subsequent text.

a) Collaborative Filtering-BORF Approach

The proposed CF-BORF utilizes a variant of the *CF* approach and employs the weighted sum method to implement scalar optimization. The Algorithm 1 illustrates the proposed CF-based approach.

1. Initialization (Line 1):

- The algorithm takes the input parameters: (a) current user identification that generates a recommendation query and (b) geographical region where the current user is currently located. 2. *Aggregate utility function computation (Line 2–Line 7):*
- The aggregated utility function computes the users' similarity in terms of venue preferences and calculates the user-to-venue proximity score by

utilizing (10). The function $computsimset(\,)$ computes the edge weights of the current user c with the expert users by utilizing the similarity formula described in (5). In Line 5, the function $computsimD(\,)$ collects those venues of the expert user that are in the closest proximity of the current user. The aggregate similarity of the current user with the neighbor (expert) users is computed in the Line 6 by utilizing function $computeagg(\,)$. The computation of aggregate similarity is performed using the following equation:

$$s_{agg}(\mathrm{s},\varsigma) = \frac{\tau}{\tau+\gamma} \times s_r(\mathrm{s},\varsigma) + \frac{\gamma}{\tau+\gamma} \times s_{clo}(\mathrm{s},c'_v),$$
$$\text{where} \quad \tau = \frac{s_r\,(\mathrm{s},\varsigma)}{\sum_{k=1}^{n} s_r\,(\mathrm{s},\varsigma'_k)},$$
$$\text{and} \quad \gamma = \frac{s_{clo}\,(\mathrm{s},c'_v)}{\sum_{k=1}^{n} s_{clo}\,(\mathrm{s},c'_{v_k})}, \tag{10}$$
$$s.t{:}\ \sum_{k=1}^{n} s_r\,(\mathrm{s},\varsigma'_k) \text{ and } \sum_{k=1}^{n} s_{clo}\,(\mathrm{s},c'_{v_k}) \neq 0.$$

- Here, s_{agg} function indicates the overall aggregate similarity with respect to preferred venues and user-to-venue closeness. The user's similarity in terms of preferences is scaled by the average of users' similarity in a specific region denoted by parameter γ. The user-to-venue closeness is scaled by the average of user-to-venue closeness, and is indicated by the parameter τ.

3. Recommendation module (Line 8–Line 9):

- On completion of the N number of iterations, the algorithm generates the top-N venues for the user by applying the traditional CF-based recommendation formula [13] stated as follows.

$$r_{c,v} = \bar{r_c} + \sum_{c' \in S_r} s_{agg}(\mathrm{s},\varsigma) \times \left(r_{c',v} - \bar{r}_{c'}\right), \tag{11}$$

where $r_{c',v}$ is the number of check-ins of user c' at any specific venue v. Parameter $\bar{r_c}$ and $\bar{r}_{c'}$ are the average number of check-ins performed by user c and c', respectively. The parameter s_r in (11) denotes the set of users that are most similar to the user c and who have also performed check-in at venue v.

b) Greedy-BORF Approach

We propose a greedy approach that generates a set of top-N venue recommendations by traversing a graph of the expert users. The basic motivation is to extract suitable venues from a network of like-minded people who share the similar preferences for various venues they visit in a geographical region. The proposed approach assigns an initial weight on the links among nodes in the graph of expert users. Subsequently, the venues are recommended by those users that are not only the most similar to the current user, but also provide maximum contribution of the venues that needs to be recommended to the current user. In this way, the Greedy-BORF approach finds an optimal path on the graph that carries a collective opinion about venues by a group of expert users. Algorithm 2 illustrates the step-by step procedure of the greedy-BORF approach for online recommendations. .

1. Initializations (Line 1–Line 4):

- The identification and geographic location of the current user is taken as the input of the Algorithm 2.
- In the Line 2, the similarity graph of the expert users is retrieved, shown in Fig. 2. The parameter $\mathrm{w(c,e)}$ in Fig. 2 represents the weight of the link in the similarity graph between the root node c and the expert user e. To compute w(c,e), the edge weights between two nodes in different levels of the graph are multiplied, and then divided by the number of edges between the two nodes. As reflected in Fig. 2, the weights of the links at level 1 of the graph (at an edge distance of one ($\delta_{\mathrm{sj}} = 1$)) are assigned according to the non-zero similarity between the current user and the expert users. Therefore, in Line 3 of Algorithm 2, only those neighbors of current user are selected from the graph that have non-zero similarity computation with the current user. In Line 4, the current user node is stored in the list known as *visitedlist.* From here onwards, we interchangeably refer to the expert users as neighbor nodes.

2. Iterative solution construction (Line 5–Line 22):

- In the Line 5, the neighbor nodes ($\mathrm{K_a}$) are sorted in the descending order based on the similarity that is further multiplied by the 1/edge distance between the current user and neighboring node (denoted as $\eta(i,j)$).
- In the Line 6–Line 10, only those venues are selected from the neighboring nodes that were not previously visited by the current user (Line 7).The selected venues are appended in the matrix M (Line 8). The visited neighbor is stored in the *visitedlist* (Line 9).
- If at Line 11, the venue count in the matrix M is greater than the required number of venues N, then the control jumps to Line 22 that computes the geographical distances of the venues in the matrix M from the root node (current user). However, if the required venue count is not achieved, then the control jumps to Line 14, where a node is selected amongst the neighbor set (K_a). The criterion for the node selection is that there should be maximum link weight from current user to the selected node, and the selected node has the maximum number of venues available for the current user. If no such node is found, then this means that the terminal node of the graph has been reached. Subsequently, the control will jump to Line 22. Otherwise, the selected node will be set as a new temporary current user (c) (Line 14). Moreover, the edge count will also be incremented by one in Line 18. From Line 20, the control will jump back to Line 6, and the procedure will be repeated iteratively.

Aggregate venues provided by the best nodes (Line 23):

Algorithm 2. Greedy-BORF approach for Venue Recommendation

Input: Current user: s, Type: $\mathcal{C}$, region: R

Output: A set V' of top-N venues visited by expert user similar to current user.

Definitions *K_j= neighbor set of node j, δ_{ij} = edge count between i and j, $\eta(i,j) = 1/\delta_{ij}$ and, Z_j= number of required venues found at a node j, visited list=* $\emptyset$.

1: $a \leftarrow c$; $\delta \leftarrow 1$;
2: $G_c \leftarrow getSimGraph(\mathcal{C}, R)$
3: $K_a \leftarrow \{x: G_c | sim(a,x) > 0\}$
4: $visitedlist \leftarrow a$
5: Sort K_a in terms of $[Sim\ (a,j) \times \eta(i,j)]$, $j \in K_a$ (descending)
6: **for** each $e \in K_a$ **do**
7: $S \leftarrow \{v: V_e | v \notin V_a\}$
8: $M \leftarrow M.append(e, S)$
9: $visitedlist \leftarrow visitedlist \cup \{e\}$
10: **end for**
11: **if** $venueCount(M) \geq N$ **then**
12: go to Line 23
13: **else**
14: $\forall j \in K_a$, set $a \leftarrow j$, such that we have $arg\,max\left[Sim(a,j) \times \eta(i,j) \times \frac{Z_j}{N}\right] \wedge\ K_j \neq \emptyset \wedge\ \forall g \in K_j |\ g \notin visitedlist$
15: **if** $No\ any\ such\ node\ found\ in\ Step\ 15$ **then**
16: go to Line 22
17: **else**
18: $\delta \leftarrow \delta + 1$;
19: go to Line 6
20: **end if**
21: **end if**
22: $D' \leftarrow computeDist(l_c, M)$
23: $V' = aggregateranking(M, D')$
24: **return** V'

- The venues are ranked and sorted in the descending order to generate top-N venues to be recommended to the current user. The following equation is used to rank the venues.

$$Rank_x = \frac{\sum_{e \in visitedlist} w(c,e) \times \eta(i,j) \times d(c,x) \times r_{ex}}{\sum_{e \in visitedlist} w(c,e)}. \tag{12}$$

Here, x is the venue to be ranked, the parameter c is the current user node, and r_{ex} is the number of check-ins performed by the expert user e at venue x. The parameter $w(c,e)$ represents the weight of the link in the similarity graph between the root node c and the expert user e. The parameters $d(c,x)$ represents 1/geographical distance between the root node c and the venue x. Equation (12) provides optimal ranking as it simultaneously considers the factors, such as edge weight among users, edge distance, geographic distance, and the amount of check-ins performed by the users at venues while computing the ranking. Next we present an example to illustrate the Algorithm 2.

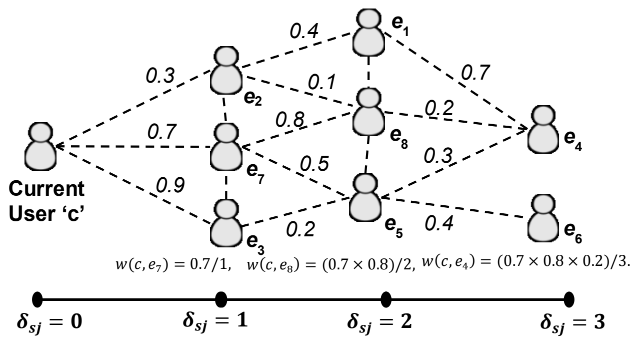


Fig. 2: Current user's similarity graph with the expert users.

An Illustrative Example: For this example, we will make use of Fig. 2 and Table 2 to illustrate working of Algorithm 2. However, it is important to note here that there is no direct data connection between Fig. 2 and Table 2, and they are just discussed together in the example to illustrate the flow of Algorithm 2. Suppose we want to recommend ten venues to the current user. The entries in the Table 2 are the check-ins performed by the expert users at a particular venue, whereas the last column of Table 2 reflects the total number of venues visited by an expert user out of the required ten venues (to be recommended). The Line 6–Line 10 of Algorithm 2 collects the venues from the neighbors (e_3, e_7, and e_2 in Fig. 2) of the current user *c* by making use of Table 2. The number of venues collected from each of the neighbor are e_3(1), e_7(3), and e_2(2). It can be observed from Fig. 2 that e_3 has highest similarity with current user, but it has least number of venues to contribute for recommendation. Therefore, to get an optimal solution, we cannot simply rely on similarity, but the number of venues contributed by each expert user must also be taken into account. The control jumps to Line 14 of Algorithm 2, where the best of the visited neighbor nodes is selected based on the similarity, and the number of venues contributed by the neighbor node. Consequently, we get the following values for each of the neighbor: e_3[0.9×1×(1/10)=0.09], e_7[0.7×1×(3/10)=0.21], and e_2[0.3×1×(2/10)=0.06]. Here, e_7 will be selected for next level traversal as the new root node (*c*) because of having the highest value. Line 6-Line 10 of Algorithm 2 will be executed again and the venues collected from the expert users will be $e_1(0)$, $e_8(2)$, and $e_5(3)$ (see Fig. 2 and Table 2). On the execution of the Line 14 of the Algorithm 2, the following values will be collected: e_8[0.8×(1/2)×(2/10)=0.08] and e_5[0.5×(1/2)×(3/10)=0.075]. Therefore, e_8 will be selected for next level traversal in the graph. As the Line 6–Line 10 are again executed, the venues collected from the neighbors are $e_4(3)$. The node e_4 does not have any further neighbors. Therefore, the condition of the Line 15 will become true and the execution continues to Line 22 where geographic distances are computed and final ranking is generated in Line 23 by using (12).

3.2.2. Vector Optimization

We present a novel bi-objective vector optimization method termed as GA-BORF by utilizing existing state-of-the-art evolutionary algorithm NSGA-II [16]. The vector-based approach optimizes all objectives (represented as vectors) simultaneously in such a manner that a solution cannot improve in one objective without compromising the other objective [15], [16]. We selected NSGA-II because of its widespread popularity in solving multi-objective optimization problems [16]. It has been shown previously that NSGA-II is able to find better spread of solutions, and better convergence near optimal solution, with low complexity compared to many other counterpart algorithms [16]. The NSGA-II algorithm [16] suggests optimal top-*N* recommendations and is divided into two phases: (a) recommendation generation and (b) recommendation optimization.

The recommendation generation phase uses the CF method with *confidence measure* (as described in the Subsection 3.1.2) to find out preferred recommended venues. The recommendation optimization phase takes the recommended venues as an input and performs optimization based on the preferred location and venue closeness using the NSGA-II [16]. The NSGA-II presents a set of the candidate solutions called a population. The population of individuals evolves towards the better solutions by employing the genetic operators, such as selection, mutation, and crossover [11], [16]. In our scenario, each of the individual is defined as a sequence of the top-*N* recommendation list $[r_1, r_2, r_3, \ldots, r_n]$. Every single element in the list of recommended venue is termed as a gene. Moreover, every gene in the list (*top-N* recommendation) consists of: (a) venue identification number and (b) location of the venue (GPS coordinates). Algorithm 3 presents the step-by-step description of NSGA-II.

1. Initializations (Line 1– Line 2)

- The multiple solutions for a user in the form of suggested recommendations are a list of inputs for the NSGA-II algorithm. In the proposed framework, the recommended venues are arranged in *top-N* ascending order. Therefore, we selected permutation-based encoding technique [11] to generate population of individuals.

TABLE 2
Number of times required venues are visited by each expert user and total check-ins at the venues

	v_1	v_3	v_4	v_7	v_{11}	v_{12}	v_{22}	v_{25}	v_{44}	v_{45}	z_j
e_1	-	-	-	-	-	-	-	-	-	-	**0**
e_2	1	-	-	-	-	-	-	-	8	-	**2**
e_3	-	7	-	-	-	-	-	-	-	-	**1**
e_4	53	3	-	9	-	-	-	-	-	-	**3**
e_5	-	27	13	45	-	-	-	-	-	-	**3**
e_6	-	-	-	-	-	-	-	41	-	29	**2**
e_7	-	-	15	-	-	12	16	-	-	-	**3**
e_8	-	-	-	-	13	-	20	-	-	-	**2**

Algorithm 3. NSGA-II based Venue Selection

Input: *R:* set of recommendations.

Output: top-*N* Recommendations based on bi-objective optimization.

Definitions: *Pop=set of population, Epop=set of population after evaluation, gen= number of generations,* Q_t *= Set of top-N optimized recommended venues,* p_{size}*= total size of population.*

1: $parents \leftarrow 0$; $f_L \leftarrow 0$;
2: $Pop \leftarrow randpop\ (p_{size},\ R)$
3: $Epop \leftarrow evaluate\ (Pop)$
4: $PP \leftarrow nondominsort\ (Epop)$
5: $S \leftarrow selectParent\ (PP,\ p_{size})$
6: $Q_t \leftarrow crossoverMut\ (S,\ p_{cros},\ p_{mut})$
7: **while** $(gen \leq max\text{-}gen)$
8: $CC \leftarrow evalute\ (Q_t\ )$
9: $R_t \leftarrow PP\ \cup Q_t$
10: $F \leftarrow nondominsort\ (R_t)$
11: **for** *each* $f_i \in F$ **do**
12: $CDA\ \leftarrow cda(f_i)$
13: **if** $size(parent) >\ p_{size}$
14: $f_L\ \leftarrow i$
15: **else**
16: $parents \leftarrow parents\ \cup\ f_i$
17: **end if**
18: **end for**
19: **if** $size\ (parents) < p_{size}$ **then**
20: $f_L\ \leftarrow ccf\ (f_L)$
21: $parents \leftarrow parents\ \cup\ f_L$
22: **end if**
23: $S\ \leftarrow selectParent(parent,\ p_{size})$
24: $pop\ \leftarrow Q_t$
25: $Q_t\ \leftarrow\ crossoverMut\ (S,\ p_{cros},\ p_{mut})$
26: **end while**
27: **return** Q_t

2. Evaluation-based on Objective Functions (Line 3)

- In Line 3, the performance of every single individual of the population is evaluated based on the fitness functions.
- The fitness function aims to compute the problem specific user defined heuristic [17]. The fitness function computes the ranking score of each recommended venue associated with an individual where the venues' ranks were computed by the HA inference. The fitness function f_1 of an individual in a population is computed as follows:

$$f_1 = \frac{\sum_{i=1}^{n}(ranked\ venues)_i}{t}, \tag{13}$$

where the parameter t represents a total number of genes in a single individual. The second fitness function f_2 computes the geospatial distance between the current user's location and the venue of each of the corresponding gene of an individual as follows:

$$f_2 = \frac{1}{\sum_{i=1}^{n} cost\ (l_u, v)_i \times t}. \tag{14}$$

- The parameter n represents the total length of an individual. The inverse of the aggregated sum of the cost function $cost(v_d, l_u)$ calculates the geospatial closeness between the current location of the user l_u and the consecutive venues v (genes) of the subsequent *ith* individuals. The user-to-venue geospatial distance is calculated using Haversine formula described in Subsection 3.1.2. The fitness function f_2 provides the overall fitness for the venue closeness of a single individual in a population.

3. Selection (Line 4–Line 5)

- As specified by NSGA-II, a non-dominated sorting approach is used to classify the entire population. The function $nondominsort$ ( ) acquires an individual (set of recommendations) from the population that is non-dominated from the rest of population. For instance, consider a set of individuals in a population $P = \{i_1, i_2, \dots, i_n\}$. Each individual is assigned fitness functions f_x and f_y. According to non-dominated sorting algorithm, in case of bi-objective optimization, the individual i_c dominates the individual i_{c+1} if and only if:

$$(f_x(i_c) > f_x\ (i_{c+1})) \text{ and } f_y(i_c) \geq f_y\ (i_{c+1}) \tag{15}$$
or
$$(f_x(i_c) \geq f_x\ (i_{c+1})) \text{ and } f_y(i_c) > f_y\ (i_{c+1})$$

- According to NSGA-II, all the individuals in a population are sorted based on (15). Intuitively, the non-dominated set of lowest level will have the highest priority to be a candidate parent for the next population in Line 4.

4. Generate intermediary population (Line 6)

- The intermediary population is generated by applying the genetic operators, such as crossover and mutation from the set of the parent individuals selected in the Line 6.
- In BORF, the sequence of recommendation is important because the top most recommended venue is the highly preferred one for the current user. Therefore, we utilized an ordered crossover method that selects venues from the parent individuals.
- Mutation operator is a genetic operator that maintains the genetic diversity from one generation of the population of individuals in the next generation [16]. We select the swap mutation operator that is commonly used in a permutation-based representation [11]. Swap mutation generates individuals by randomly swapping two genes from the individual [11].

5. Iterative procedure for generating best solutions (Line 7–Line 26)

- The Line 7 evaluates the offspring (q_t) depending on the fitness functions described in (13) and (14). The Line 8 will generate a merged population (R_t) of individual candidates through combining the population of parents and the offspring of size *2N*.
- The overall population is of size *2N*. Therefore, all the individuals that are categorized as a lower-to-higher level using a non-dominated sorted algorithm cannot be accommodated in a new population of size *N*. To accommodate new size *N* population (Line 12), the Crowded Distance Assignment (CDA) [16] is calculated. The CDA basically estimates the density of the individuals with respect to the neighboring individuals. The CDA will be further used in the Crowded Comparison Function (CCF) [16], described in the subsequent text.1
- To accommodate exactly the *N* number of population, the individuals that are arranged level-wise are compared. If the number of individuals in a level is less than the total population size of *N*, then the current level will be selected for the next generation (Line 16). However, if the number of individuals in a level is larger than the population size *N* then the last level individuals are organized using the CCF (Line 20) [16].
- Finally, the best individuals from the merged population undergo a crossover and mutation and are combined with the original population to form a new population of the candidate individuals (Line 23–Line 25). If the number of generations is less than the maximum number of required generations, then the algorithm will perform iteration. Otherwise, the population of optimal individuals in a form of the top-*N* recommendations will be generated for a specific region *R* (Line 27).

3.3. Time Complexity Analysis

In this subsection, we compute the time complexity of the pre-processing phase, CF-BORF, the greedy-BORF, and GA-BORF approach, respectively. Moreover, we present the performance evaluation of the proposed BORF. For time complexity analysis, in a specific number of regions, the time complexity of the HA inference model is $O\ (a \times r \times (x'^2 + y^2)$, where the parameter a presents the total number of iterations for approaching to the convergence, x' and y present total number of users and the venues in a region r. The time complexity of the similarity computation for an expert user is $O(r \times x^2)$ and the proximity computation graph is $O(r \times x \times y)$. The total time complexity of HA, ranking and mapping computation is $O\ (r \times \left(\left(a \times (x'^2 + y^2)\right) + y \log y\right)$. For the higher values of venues y, the value of $y \log y$ become insignificant. Therefore, the overall time complexity of the offline pre-processing phase would be $O\ (r \times \left(\left(a \times (x'^2 + y^2)\right)\right)$.

The time complexity of Line 2–Line 7 of the CF-BORF is $O(\ x \times y^2)$. The Line 8 has an overall complexity of $O\ (x)$. We added all the complexities as $O\ [top_N \times \left((x \times y^2) + x\right)]$. The value of the top-*N* is smaller than the total number of expert users and the popular venues. Therefore, the overall complexity of CF-based bi-objective optimization algorithm is $O\ (x\ \times y^2)$.

The time complexity of the greedy-BORF computes the similarity with the set of expert users. The similarity function for y venues is $O(y)$.Therefore, total time complexity of Line 2 for x expert users is $O(x+y)$. The line 5 takes $O(x + log\ x)$ to sort the x experts. In the worst case, the Line 6- Line 10, number of iterations is x, and the Line 5- Line 14 also takes x iterations. The time complexity of Line 14 is $O(x)$. The combine time complexity of Line 5- Line 14 is $O(3y^2 + x\ log\ x)$. The time complexity of Line 22 is $O(n)$, where n represents the number of venues that can be recommended and $O(x \times n)$ is the time complexity of Line 23. Therefore, the time complexity for Algorithm 1 for 1 region is $O(x^2 + x\ (log\ x + n)\ )$. For r regions, the time complexity becomes $O(r(x^2 + x\ (log\ x + n)\ ))$.

In the GA-BORF approach, the time complexity of NSGA-II in Line 1 is $O\ (y^2)$ because of the process of generating the random population of the top-N recommendations of size N in a region. The Line 2 evaluates each individual with respect of objective functions. The time complexity of evaluation function is $O\ (\ M\ (y^2 \times x^2))$, where parameter M is the number of objective functions. To identify the individuals related to a first non-dominated rank in a non-dominated sorting algorithm. Every single individual is compared with the other individual to find the dominance with a complexity of $O\ (M\ (y\ \times x))$. For the multiple iterations to find out all the dominated solutions, the total complexity of the non-dominated sorting algorithm is $O\ (M\ (y^2 \times x^2))$. The complexity of a crowded distance is $O(x + y\log y)$. We conglomerate the overall time complexity of the NSGA-II-based recommendation algorithm to be as $O\ (x\ log\ x)$.

4 Performance Evaluation

We compare our results with the following related schemes: (a) User-based Collaborative Filtering [7] (UCF), (b) Matrix Factorization (MF) [10], and (c) Random Walk with Restart (RWR) [6]. For all the aforementioned schemes, the user-to-venue closeness information is also taken into account, as the user-to-venue closeness is computed after a list of venues are generated by a baseline approach. A brief description of the schemes is presented in the next subsection.

4.1 Baseline Techniques

- UCF computes similar users who visited the similar venues in the past are most likely visit the same venues in the future [1], [2].
- The MF approach maps the users and venues to a joint latent factor space of a dimensionality a. A user x is related to a row vector $p_x \in R^a$ and a venue y is associated with a column vector $q_y \in R^a$. The estimated rating of the user x for a venue y can be stated as $r_{x,y} = p_x^T \times q_y$, where $r_{x,y}$ estimates the user's overall interest in a particular venue in VRS.
- The RWR method combines the data about frequently visited venues and the friends, represented as social ties in a graph using a structured transition matrix [6]. The RWR leverages several sources of the data and encode them into a network structure. The RWR performs a personalized random walk on the graph with a restart to suggest the recommendations for an individual user.

4.2 Results

We utilized *"Gowalla"* dataset consists of 6,442,890 check-ins performed by 150,734 users in total number of 1,280,969 venues [6]. We performed extensive experiments on our internal OpenNebula cloud setup running on 96 core Supermicro SuperServer SYS-7047GR-TRF systems. In the selected dataset, out of the entire records, 80% of the record is used as the training set and 20% constitute the test set for the evaluation. For each data point, we performed 25 independent runs. We used a standard 5-fold cross validation technique for evaluating the accuracy rate of the framework [2].

We utilized the three standard performance evaluation matric to evaluate the proposed recommendation frameworks: (a) precision, (b) recall, and (c) F-measure [26]. The precision presents a ratio of the accurate recommendations (true positive (tp)) to the total number of anticipated recommendations (tp+ false positive (fp)). An accurate recommendation is the recommendation that has been predicted correctly in the top-N recommended venues. Precision is given as:

$$Precision = \frac{tp}{tp + fp}. \tag{16}$$

The recall measures the single user recommendation effectiveness by computing the average quality of the individual recommendations. Recall is defined as the ratio of correct recommendations (tp) to the total number of recommendations (tp + fn). The recall presents the proportion of all the accurate recommendations in the top-N recommended venues and can be represented as:

$$Recall = \frac{tp}{tp + fn}. \tag{17}$$

The F-measure is the harmonic mean of precision and recall and is denoted as follows:

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{18}$$

Fig. 3 (a), (b), and (c) present the precision, recall, and f-measure results without incorporating the pre-processing phase. Whereas, Fig. 3(d), (e), and (f) show the precision, recall and f-measure results with pre-processing phase. As reflected in Fig. 3, the results in (d), (e), (f) show better performance in terms of precision, recall, and f-measure as compared to the results in Fig. 3(a), (b), (c). Such improvement in results is due to the fact that the pre-processing phase reduces the negative effect of data sparseness over recommendation quality. Data sparseness results in zero similarity values in collaborative filtering, and with large number of zero

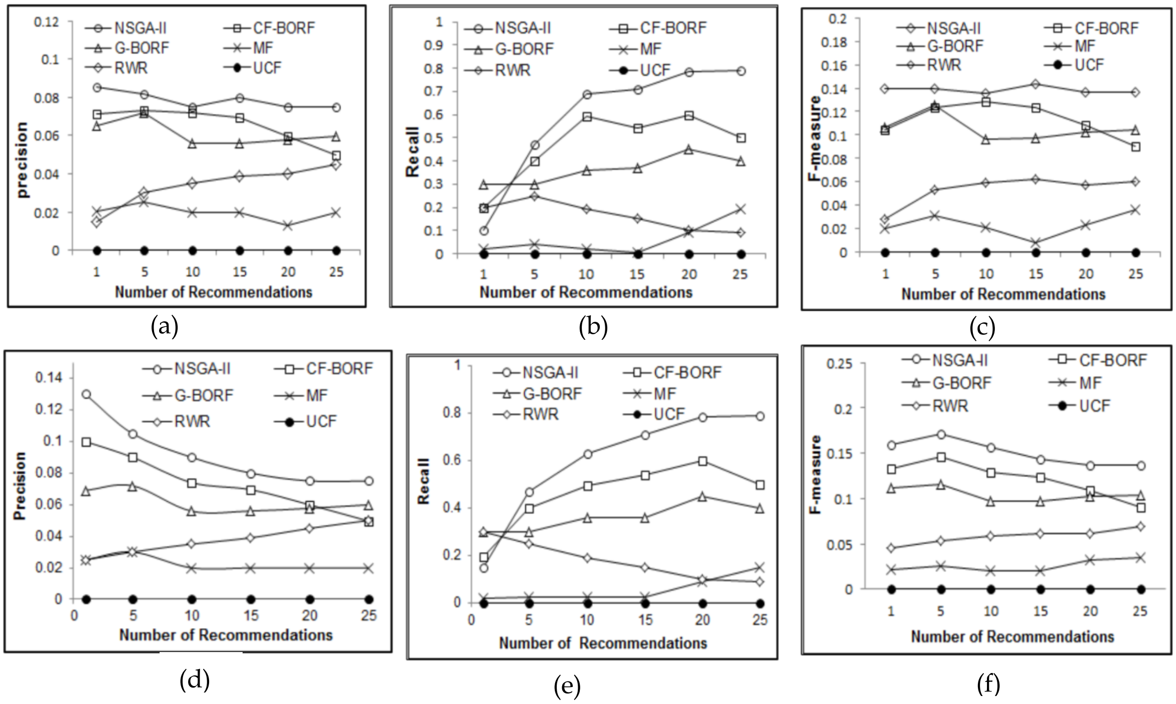


Fig. 3. Performance evaluation results without pre-processing: (a) Precision, (b) Recall, and (c) F-measure, and with pre-processing: (d) Precision, (e) Recall, (f) F-measure.

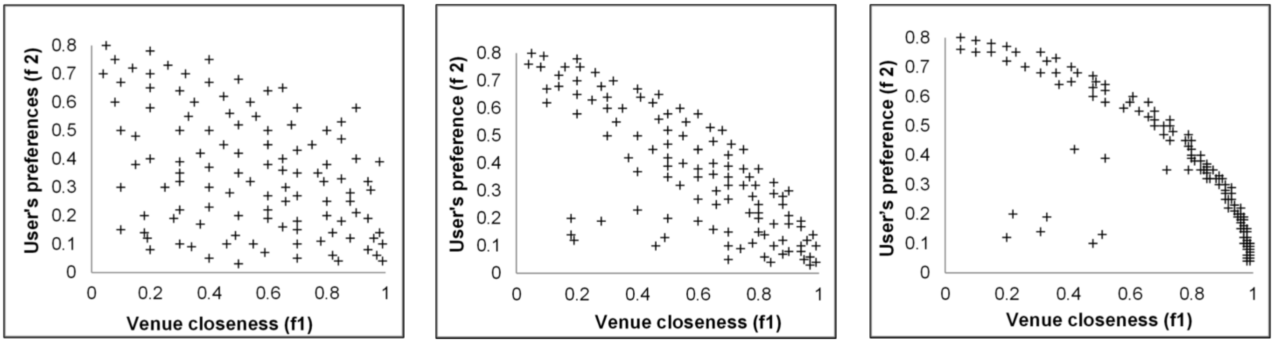
entries in user-to-user similarity matrix the recommendation quality decreases. Despite the fact that any two persons have visited almost the same set of venues, the similarity value of the two persons will be smaller or zero if they have significant difference in visit patterns. To reduce the number of zero entries in user-to-user weighted matrix in the aforementioned scenario, we augmented similarity values with confidence measure (or conditional probabilities). In that way, if similarity of two persons is zero but they have visited almost similar set of venues (with different patterns), then they will not be assigned a zero weight in the user-to-user matrix, which overall improves the recommendation quality.

As reflected in Fig. 3(d), (e), and (f), NSGA-II demonstrates the better performance in terms of precision and recall as compared to the rest of the schemes (CF-BORF and greedy-BORF). In contrast, the CF-BORF and greedy-BORF approaches present slightly lower performance because of the aggregation method that maps the users' preferences and location closeness into single objective function. Such aggregation cannot provide accurate results especially when there is tradeoff between the user's preferences and location closeness. For instance, in the case of CF-BORF, when there is no similarity between two users' preferred locations, the venue will be suggested to the current user on the bases of user-to-venue closeness. Such suggestion may not provide optimal recommendation and indicates lower performance in terms of precision and recall as presented in Fig. 3(a) and (b).

We compare the proposed optimization techniques (CF-BORF, greedy-BORF, and GA-BORF) for venue recommendation with the existing UCF, MF, and RWR techniques. As reflected in Fig. 3(a), CF-BORF, greedy-BORF, and GA-BORF present the better performance in terms of precision and recall as compared to the rest of the existing schemes, such as UCF, MF, and RWR. The improved performance is because the proposed techniques optimize the recommendation by taking into account the user preferences based on similarity computation and user-venue closeness. The venue suggestions based on such optimization are not only the most preferable for a given user, but also located in the closest proximity of a user's current location. The application of confidence measure and HA inference model effectively helps to obtain better solution that results in an increased recommendation precision.

The RWR method demonstrates high performance in terms of precision and recall as compared to the traditional CF-based approach, such as MF and UCF. The reason is that the RWR does not compute the similarity by utilizing user-to-user similarity matrix. Therefore, RWR is not significantly affected by data sparseness issues. The UCF indicates very low performance in terms of precision and fails to provide any significant results due to highly sparse dataset of "Gowalla". The tradeoff between precision and recall is depicted in Fig. 3(b). Compared to other schemes, the GA-BORF indicates better performance in terms of the F-measure as presented in Fig. 3(c).

A series of simulation runs were conducted to test the effectiveness of the NSGA-II algorithm utilized in GA-BORF method. Top-*N* venue recommendation problem is formulated as a multi-objective optimization problem, and NSGA-II is applied to simultaneously maximize both the objectives, such as user's preferences and venue closeness. Permutation-based encoding technique and ordered crossover method are used for the population



(a) (b) (c)

Fig.4. Multi-objective performance measure for NSGA-II (a) Generation size 5, (b) Generation size 100, and (c) Generation size 200

generation. The NSGA-II reports best performance for crossover rate equal to 0.9 and mutation rate equal to 0.1, respectively. These parameter values were determined empirically through numerous runs on "Gowalla" datasets. The maximum number of generation size has been selected as 0 and 400, respectively.

Figs. 4(a), 4(b), and 4(c) present the impact of user's preferences (vertical axis) and venue closeness (horizontal axis) on the recommendations. Numerous simulations are performed to show the effect of generation size on NSGA-II's performance. It can be observed from Fig. 4(a) that with the generation size of 5, the solutions are not converging, which indicates small generation size does not yield good recommendations. For generation size of 100 (Fig. 4(b)), the solutions appear to converge slightly, thereby improving recommendation quality. Fig. 4(c) presents the performance of NSGA-II by increasing the number of generations to 200, which shows the maximum convergence with improved solution quality in terms of recommendations. Moreover, in Fig. 4(c), the convergence of non-dominated solutions toward the optimization of both objectives is apparent. The convergence curve [11] depicted by Fig. 4(c) contains the better spread and solutions by considering the two objectives (users' preferences and location closeness).

Table 3 presents performance of both objective functions in terms of precision and recall. The NSGA-II demonstrates better performance as we increase the number of generation size. For instance, in case of generation size of 300, the precision of recommendations is greater than 0.02. Similarly, in case of 400 generation size the precision is greater than 0.08. The improved performance is because of the fact that the NSGA-II generates the optimal recommendations by maximizing both the objective functions simultaneously. Therefore, only those venues are recommended to the current users that are not only similar to user's preferences but also located in the closest proximity of user's current location. Increasing the number of generation size gradually prune all the dominated solutions. Consequently, remaining non-dominated optimal solutions comprised of the recommendations that are the best suggestions in the trade-off between the venue preferences and venue closeness.

4.3 Significance Testing

We utilized the paired t-test [19] to evaluate statistical significance among the algorithms. We used the null hypothesis for algorithmic comparisons, which illustrates that the true mean difference between the algorithms is zero. The zero mean difference shows that the tested algorithms are significantly related. The p-score in paired t-test ranges from 0 to 1. The p-score value closer to 0 means the algorithms are significantly related. Alternatively, the p-score value closer to 1 means that the tested algorithms are significantly different. The paired t-test between NSGA-II (5 generations and 100 generations) for accuracy values yielded the average difference: 0.0913, standard deviation: 0.2007, standard error of mean difference: 0.0186, t-score: 0.4901, and p-score: 0.6875. Moreover, paired t-test between NSGA-II (100 generations and 200 generations) for accuracy presents

TABLE 3
THE EXPERIMENTAL RESULTS FOR NSGA-II

No. of generations	f(1)	f(2)	Precision	Recall
100	0.1	0.5	0.0211	0.0234
150	0.2	0.3	0.0227	0.090
200	0.4	0.1	0.0347	0.042
250	0.5	0.6	0.0434	0.012
300	0.6	0.1	0.0738	0.077
350	0.5	0.9	0.092	0.028
400	0.9	0.7	0.1391	0.095

TABLE 4
P-SCORE OF NSGA-II WITH OTHER ALGORITHMS

Algorithm	p-score
CF-BORF	0.96
G-BORF	0.97
MF	0.99
RWR	0.97
UCF	0.97

the average difference: 0.0023, standard deviation: 0.1165, standard error of mean difference: 0.0108, t-score: 0.2151, and p-score: 0.5849. The Table 4 presents the p-score of NSGA-II compared with other algorithms. It can be observed from the Table 4 that NSGA-II shows high statistical difference and improvement in terms of precision compared to the other algorithms.

5 Related Work

In the past, most work focused on trajectory-based approaches for venue recommendation systems [1]–[3]. The trajectory based approaches record information about a user's visit pattern (in the form of GPS coordinates) to various locations, the routes taken, and dwell times. The authors in [3] applied data mining and machine learning on trajectory data to recommend most popular places. Although, trajectory-based approaches recommend locations to users based on their past trajectories, a major drawback of such approaches is that they are unable to simultaneously consider other influential factors apart from simple GPS trace that makes them produce less optimal recommendations. To address such deficiency, we utilized multi-objective optimization in our proposed framework. Another issue is that the trajectory-based approaches suffer from data sparseness problem as usually a person does not frequently visits many places, which results in sparse user-venue matrix. Moreover, the trajectory based approaches suffer from scalability issues as huge volumes of trajectory data needs to be processed causing considerable overhead. Some of the approaches, such as [3], [5] are based on the online ratings provided by the users to the visited places. The authors in [7] combine the available venue ratings with users' social ties to recommend venues that are high-ranked as well as most preferred by a user's friends. However, the authors did not compare their approach with any of the baseline approaches, and does not discuss complexity of their work. The aforementioned approaches perform different modeling to users' preferences, but they are not considering multiple objectives that we specifically considered in our study. Moreover, they also suffer from data sparseness issues due to limited number of entries within the user-rating matrix.

Apart from rating based approaches, few of the techniques have their models built on check-in based approaches where the users provide small feedbacks as check-ins about the places they visited [2]–[4], [7], [14]. For example, the authors in [6] applied random-walk-with-restart on a user-venue check-in matrix to generate personalized recommendations. Most of the above mentioned approaches have their designs built on memory-based CF that enables such approaches to provide recommendations to users on the basis of their past entries. However, such approaches suffer from common drawbacks of memory-based CF (e.g. cold start and data sparsity) which reduce their performance. Moreover, large number of similarity computations on user-to-venue matrix makes such approaches less scalable. There has been some limited work performed on applying multi-objective optimization on recommendation systems. One such contribution is by Ribeiro *et al.* [15] where authors performed a weighted combination of numerous recommendation algorithms and applied optimization to find appropriate weights for the constituent algorithms. However, their approach is computation intensive and no time complexity was discussed.

To address the issues cited above, we proposed a hybrid approach over a cloud architecture that combines the benefits of memory-based and model-based collaborative filtering along with multi-objective optimization to obtain an optimal list of venues to be recommended. Moreover, our proposed framework presents a solution for scalability, data sparseness, and cold start issues.

6 Conclusions

We proposed a cloud-based framework *MobiContext* that produces optimized recommendations by simultaneously considering the trade-offs among real-world physical factors, such as person's geographical location and location closeness. The significance and novelty of the proposed framework is the adaptation of collaborative filtering and bi-objective optimization approaches, such as scalar and vector. In our proposed approach, data sparseness issue is addressed by integrating the user-to-user similarity computation with confidence measure that quantifies the amount of similar interest indicated by the two users in the venues commonly visited by both of them. Moreover, a solution to cold start issue is discussed by introducing the HA inference model that assigns ranking to the users and has a precompiled set of popular unvisited venues that can be recommended to the new user.

In the future, we would like to extend our work by incorporating more contextual information in the form of objective functions, such as the check-in time, users' profiles, and interests, in our proposed framework. Moreover, we intend to integrate other approaches, such as machine learning, text mining, and artificial neural networks to refine our existing framework.

References

[1] A. Majid, L. Chen, G. Chen, H. Turab, I. Hussain, and J. Woodward, "A Context-aware Personalized Travel Recommendation System based on Geo-tagged Social Media Data Mining," *International Journal of Geographical Information Science*, pp. 662-684, 2013.

[2] M. Ye, P. Yin, and W. Lee, "Location recommendation for location-based social networks," In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, pp. 458-461, 2010.

[3] Y. Zheng, L. Zhang, X. Xie, and W.Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," In *Proceedings of the 18th international conference on World wide web*, ACM, pp. 791-800, 2009.

[4] C. Chow, J. Bao, and M. Mokbel, "Towards Location-Based Social Networking Services," *In Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, ACM, pp. 31-38, 2010.
[5] P. G. Campos, F. Díez, I. Cantador, "Time-aware Recommender Systems: A Comprehensive Survey and Analysis of Existing Evaluation Protocols," *User Modeling and User-Adapted Interaction*, vol. 24, no.1-2, pp. 67-119, 2014.
[6] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "A Random Walk around the City: New Venue Recommendation in Location-Based Social Networks," In *Proceedings of International Conference on Social Computing (SocialCom)*, pp.144-153, 2012.
[7] Y. Doytsher, B. Galon, and Y. Kanza, "Storing Routes in Sociospatial Networks and Supporting Social-based Route Recommendation," In *Proceedings of 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, ACM, pp. 49-56, 2011.
[8] S. Seema, and S. Alex, "Dynamic Bus Arrival Time Prediction, using GPS Data," In *Proceedings of the Nat. Conference Technological Trends (NCTT)*, pp. 193-197, 2010.
[9] B. Chandra, S. Bhaskar, "Patterned Growth Algorithm using Hub-Averaging without Pre-assigned Weights," In *Proceeding of IEEE International Conference on Systems, man, and Cybernetics* (SMC), pp.3518-3523, 2010.
[10] B. Hidasi, and D. Tikk, "Initializing Matrix Factorization Methods on Implicit Feedback Database," *Journal of Universal Computer Science*, vol. 19, no. 12, pp. 1835-1853, 2013.
[11] C. Chitra and P. Subbaraj, "A Non-dominated Sorting Genetic Algorithm for Shortest Path Routing Problem in Computer Networks," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1518-1525, 2012.
[12] Y. Wang, S. Wang, N. Stash, L. Aroyo, and G. Schreiber, "Enhancing Content-Based Recommendation with the Task Model of Classification," In *Proceedings of the Knowledge and Management*, pp. 431-440, 2010.
[13] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, "Recommender Systems Survey," *Knowledge-Based Systems,* vol. 46, pp. 109-132, 2013.
[14] J. Bao, Y. Zheng, M.F. Mokbel, "Location-based and Preference Aware Recommendation using Sparse Geo-Social Networking Data," In *Proceeding of 20th International Conference on Advances in Geographic Information Systems*, ACM New York, pp.199-208, 2012.
[15] M. Ribeiro, A. Lacerda, A. Veloso, and N. Ziviani, "Pareto-Efficient Hybridization for Multi-objective Recommender Systems," In *Proceeding of 6th ACM Conference on Recommender Systems*, pp. 19-26, 2012.
[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II," *IEEE Transaction on Evolutionary Computations*, vol. 6, no. 2, pp. 182-197, 2002.
[17] H. Nasiri, M. Maghfoori, "Multiobjective Weighted Sum Approach Model reduction by Routh-Pade approximation using Harmony Search," *Turkish Journal of Electrical Engineering and Computer Science*, vol. 21, no. 2, pp. 2283-2293, 2013.
[18] J. Abimbola, "A Non-linear Weights Selection in Weighted Sum for Convex Multi-objective Optimization," *Mathematics and Information,* vol. 27, no. 3, 2012.
[19] Paired t test. Wiley Encyclopedia of Clinical Trials, 2008.

Rizwana Irfan is currently pursuing her Ph.D. at North Dakota State University, Fargo, USA. Her research domain encompasses topics related to context-awareness in social networking and multi-objective optimization techniques in recommendation systems.

Osman Khalid is an Assistant Professor at the COMSATS Institute of information Technology. His area of research in PhD includes opportunistic networks, recommendation systems, and trust and reputation systems.

Muhammad Usman Shahid Khan is pursuing Ph.D. at the North Dakota State University, Fargo, USA.

Camelia Chira obtained a PhD degree in the area of Software Agents and Multi-agent Systems for Distributed Design Environments. Dr. Chaira received the Romanian Academy Award "Grigore Moisil" in the field of Computer Science for research results based on a collection of papers regarding nature-inspired meta-heuristics for solving optimization problems.

Rajiv Ranjan is Project Leader in the CSIRO computational informatics, Canberra.

Fan Zhang received a Ph.D. degree in Control Science and computer engineering from Tsinghua University, China. Dr. Zhang is the author/co-author of more than 25 research papers in reputable journals and a senior member of IEEE.

Samee U. Khan is Associate Professor at the North Dakota State University, Fargo, ND, USA. Prof. Khan's research interests include optimization, robustness, and security of: cloud, grid, cluster and big data computing, social networks, wired and wireless networks, and optical networks.

Bharadwaj Veeravalli received a PhD degree in Engineering form Indian Institute of Science, Bangalore, India. Prof. Veeravalli is Associalte Editor of IEEE Transaction on Computers, IEEE Transaction on Systems, Man, and Cybernetics.

Keqin Li received a Ph.D. degree in Computer Science from the University of Houston, Texas, USA, in 1990. Professor Li is an Associate Editor of IEEE Transactions on Parallel and Distributed Systems.

Albert Y. Zomaya is the Chair Professor of High Performance Computing & Networking in School of Information Technologies, University of Sydney . Professor Zomaya is the author/co-author of seven books, more than 400 papers, and the editor of 12 books and 15 conference proceedings.