Cold-start Point-of-interest Recommendation through Crowdsourcing

PRAMIT MAZUMDAR, Università degli Studi Roma TRE, Italy
BIDYUT KR. PATRA and KORRA SATHYA BABU, National Institute of Technology Rourkela, India

Recommender system is a popular tool that aims to provide personalized suggestions to user about items, products, services, and so on. Recommender system has effectively been used in online social networks, especially the location-based social networks for providing suggestions for interesting places known as POIs (points-of-interest). Popular recommender systems explore historical data to learn users' preferences and, subsequently, they recommend locations to an active user. This strategy faces a major problem when a new POI or business evolves in a city. New business has no historical user experience data. Thus, a recommender system fails to gather enough knowledge about the new businesses, resulting in ignoring them during recommendations. This scenario is popularly known as a cold-start POI problem. Users never get recommendations of the new businesses in a city even though they can be relevant to a user. Also, from a business owner's perspective, such a recommendation strategy does not help its reachability among users. Therefore, it is important for a recommender system to remain updated with new businesses in a city and ensure that all relevant POIs are recommended to a user irrespective of their lifetime. A POI recommendation approach is proposed in this work that can effectively handle the new businesses, or the cold-start POI problem, in a city. We crowdsource descriptions of cold-start POIs from various online social networks. The reviews of users are exploited here to learn the inherent features at the existing POIs and the new crowdsourced POIs. Finally, the proposed approach recommends top-K POIs consisting of the existing and new POIs. We perform experiments on the real-world Yelp dataset, which is one of the largest available data resources containing details on a wide range of businesses, users, and reviews. The proposed approach is compared with four existing POI recommendation approaches. The obtained results show that our approach outperforms others in handling cold-start POIs.

CCS Concepts: • **Information systems → Recommender systems**;

Additional Key Words and Phrases: Recommender systems, crowdsourcing, clustering, Yelp network

ACM Reference format:
Pramit Mazumdar, Bidyut Kr. Patra, and Korra Sathya Babu. 2020. Cold-start Point-of-interest Recommendation through Crowdsourcing. *ACM Trans. Web* 14, 4, Article 19 (August 2020), 36 pages.
https://doi.org/10.1145/3407182

19

Authors' addresses: P. Mazumdar, Università degli Studi Roma TRE, Rome, Italy; email: pramitmazumdar@gmail.com; B. Kr. Patra (corresponding author) and K. S. Babu, National Institute of Technology Rourkela, Odisha, India; emails: {patrabk, ksathyababu}@nitrkl.ac.in.

1559-1131/2020/08-ART19 $15.00
https://doi.org/10.1145/3407182

1 INTRODUCTION

The growth of location-aware devices has led people to share their experiences about points-of-interest (POIs) through online portals. Various online services such as Foursquare, Yelp, Google places, Facebook places, and so on, provide platforms where users can share their experiences at the visited locations (POIs) themselves. Often people do it for fun, whereas in Foursquare people get rewards for sharing their experiences about various POIs. The user experiences in the form of reviews, ratings, and so on, at a location contain rich information about users and the respective POIs. Using this information, various forms of recommendations have been introduced, such as the location or POI recommendation [49, 52, 55], friend recommendation [17, 46, 54], event recommendation [18, 30, 44], and so on. Location recommendation has been a challenging task, as it depends upon a wide range of factors. A personalized location or POI recommender system aims at providing POIs that are related to the user's likings. Intuitively, a recommender system must understand a user's preferences and, subsequently, identify relevant POIs related to the learned preferences by exploiting historical information about POI and user activities. Various factors such as similar users, friends, reviews, ratings, temporal information, current location, and so on, play major roles for personalized recommendations. An important challenge for a recommender system is the cold-start problem [11]. It refers to a situation when a recommender system fails to recommend POIs due to lack of historical data. Cold-start scenarios can broadly be divided into two scenarios: user cold-start and POI cold-start. User cold-start has been addressed in many previous research works [10–12, 27, 48]. A newly evolved POI or business in a city having no historical data is termed as a cold-start POI. In the rest of this article, we refer such cold-start POIs as CSPs. A POI cold-start problem demonstrates a situation when a POI recommender system does not recommend a newly evolved POI to any user. The POI cold-start problem has not been widely explored in literature. However, working on a new POI has a three-fold advantage:

(1) It assures a recommender system with updated list of items or POIs that are relevant to a user.
(2) Users often look for new items in the recommended list, which is popularly known as novelty in recommendations. If a cold-start POI is found to be relevant to the target user, then it can be recommended to increase novelty of the recommender system.
(3) Owner of a new POI will benefit if it is recommended to users. Otherwise, it is not worthy to post an advertisement or create a webpage for the same.

In the literature of POI recommendations, the collaborative filtering (CF) approach explores only the ratings of users. Both the variants of collaborative filtering, namely, memory- and model-based CF, have been very popular in many application domains [5]. However, these methods do not explore the content information of users and POIs. Thus, they fail to handle the cold-start scenarios in POI recommendation. Cold-start scenarios are generally handled through hybrid approaches, which fuse multiple factors for recommendation. The work in Reference [52] uses a linear fusion model of user, social, and geographical influence, but the POI features are ignored. Reviews at a POI are analyzed in SELR [49] for modelling POI preferences. However, they do not incorporate newly evolved POIs into the proposed system. The CRCF model [55] divides a geographic space into regions and provides region-based recommendations by modelling features of POIs. A user who is "new" to a city is considered here as a new user at the concerned city, and such a scenario is termed as the "new city" user cold-start problem. Their approach successfully addresses the "new city" user cold-start problem, however, it fails to recommend or handle the cold-start POIs, i.e., the newly evolved POIs (CSPs).

In the present work, we propose a technique that can identify the newly evolved POIs in a city and also gather relevant information by crowdsourcing from other social networks. We term our proposed recommendation approach as a feature-based POI recommendation (FPR). The related POIs are grouped together into clusters and a representative feature set for each cluster is computed. The identified CSPs are assigned to the clusters having similar features. Thus, the CSPs along with the existing POIs are finally considered together for recommendations. Unlike the traditional techniques, we introduce a term *RecScore* that is computed for all POIs. Finally, the top-K POIs with high *RecScore* are selected for recommendation. A preliminary version of this work was presented in Young Researchers' Symposium [33]. We incorporate an aspect-based sentiment analysis technique from the presented work in Reference [33] into the current recommendation approach. However unlike Reference [33], the proposed FPR includes a feature-based clustering step, learning cold-start POIs from a higher number of social networks and also a new approach for selecting the top-K recommendations.

The contributions of the proposed POI recommender system (FPR) can be listed as below.

(1) A novel cold-start POI detection technique by crowdsourcing the online social media data is devised. It can be applied periodically for detecting and updating the recommender system database with newly evolved businesses in a city.
(2) A fuzzy technique is applied to cluster the POIs. The cold-start POIs are placed in the existing clusters based on the feature similarity. The *significance*, *popularity*, and *impact* of each feature is computed to identify the most dominating features in a cluster.
(3) We introduce the terms significance, popularity, and impact of a POI feature with respect to a cluster. Significance of a feature in a cluster captures the coverage of the target feature among the member POIs in the cluster. Whereas, popularity highlights how frequently a POI possessing the target feature has been visited by users. Thus, features for a cluster are selected by considering both its coverage among member POIs and the frequency of visit by users. Impact of a feature in a cluster is used to denote its positiveness or negativeness among the member POIs.
(4) A recommendation approach is proposed that gives equal weightage to cold-start POIs and existing POIs during the recommendation task. The geographical influence is also explored in the proposed recommender system.
(5) Experiments are performed on the publicly available Yelp dataset to evaluate the proposed recommender system in handling the cold-start POIs. The obtained results show that the proposed work outperforms existing approaches in handling cold-start POIs.

The rest of the article is organized as follows: Section 2 provides a survey on existing works related to POI recommendations. The motivation of our proposed work is described in Section 3. Section 4 describes the proposed FPR approach towards POI recommendation in detail. The experimental results are depicted in Section 5. We conclude our work in Section 6.

2 RELATED WORK

This section depicts the detailed description of recent works on POI recommender systems. The state-of-the-art works have been categorized based on the types of approaches used for recommendation.

2.1 Hybrid Approaches by Fusing Multiple Factors of LBSN for Recommendation

The collaborative filtering technique has been the most popular approach in various domains [6, 8, 14, 39]. These approaches depend upon the historical rating data available for the target users and at the target POIs. However, a cold-start POI has no historical data [24, 29, 38]. Therefore, many researchers have proposed hybrid approaches that combine collaborative and content information such as reviews, spatio-temporal data, and so on, to perform POI recommendations [2, 10, 51, 52].

An active user's visit to a location is mostly influenced by social and geospatial factors. The work in Reference [51] proposes two location recommendation techniques based on collaborative filtering: friend-based collaborative filtering (FCF) and geo-measured friend-based collaborative filtering (GM-FCF). The FCF technique considers the ratings at commonly visited locations by the social network friends. Whereas, GM-FCF extends FCF to combine a linear regression technique over the distances between the active user and its friends to compute a similarity score between them. However, both the recommendation approaches do not learn the new cold-start POI in a city for recommendation. In Reference [2], a regularized matrix factorization technique (RMF) is utilized to predict the user ratings at various POIs. In this regard, users and POIs are mapped to joint latent factors that represent user preferences for a POI. Similarly, the approach in Reference [58] generates a tree-based hierarchical graph (TBHG) with the location histories of the active users. The user preferences are learned from the TBHG by utilizing the HITS algorithm [22]. However, the approaches in References [2, 58] fail to consider the geographical influence in POI recommendations. In this regard, the proximity of the target user plays a very important role.

The user preferences, social influences, and geographical influences (USG) are combined in Reference [52] for POI recommendations in location-based social networks. A friend-based collaborative filtering is used to compute social influence of a friend on the target user. They identified existence of spatial clusters from the user check-ins of a location-based social network user. Therefore, the geographical influence of a location becomes an important factor. A naive Bayes method has been proposed that computes the probability of a target user to visit a location based on the geographical distance between them. Finally, all these three factors are combined to assign a score for each POI with respect to the target user. The top-K POIs with highest scores are selected for recommendations. A similar approach is followed in UPS-CF [10], which combines user preference, proximity, and social-based collaborative filtering for recommending out-of-town users. Usually, the user reviews play an influencing factor in selecting a place to visit. In this regard, the USG and UPS-CF do not exploit the inherent features of a POI based on the available user reviews on it.

2.2 Sentiment-based Approaches for Recommendation

Recently, sentiments of users have been exploited to recommend POIs [15, 19, 21, 47–49, 53, 57]. Yang et al. in Reference [49] analyze the tips (customer reviews) at various POIs in the LBSNs for location recommendation. The tips are processed using a dictionary-based sentiment analysis technique. A user's rating at a location is identified by aggregating the sentiment scores of each word in the corresponding review. In the rest of this article, we denote this work as SELR. The approach in SELR does not consider the inherent features of a POI. Moreover, learning both the positive and negative features of a POI helps in improving a recommendation. This aspect is not considered during the preference learning module in SELR. Zhang et al. in Reference [57] use an explicit factor model for recommendation. Here the aspects are identified as the explicit features of items and corresponding phrase-level sentiment analysis is performed for finding an understandable opinion related to a review. The user-feature, item-feature, and existing user-item ratings are exploited for recommendation. This work successfully identifies the items to be recommended, as well as items that should not be recommended for a given user. Similarly, Jin et al. in Reference [19] propose a rating prediction methodology that exploits the semantics of the reviews. The social and topic aspects of user check-ins are utilized in Reference [15] for recommendation. Reviews are explored to model the user preferences. The words in reviews are used to understand semantic meaning of POIs. Finally, the topic distribution of users and the word semantics of POIs are correlated to perform location recommendation.

The state-of-the-art in sentiment analysis techniques do not consider one typical scenario where an expression can be related to both an entity and an aspect for a certain sentiment. Yang et al. in Reference [50] proposed a multi-entity aspect–based sentiment analysis technique that can provide (entity, aspect) pairwise sentiments. However, lack of enough reviews makes it difficult to generate a POI profile with representative features. Very recently, Song et al. [43] proposed a deep memory network (CADMN) to deal with sparsity in available reviews. These recent advancements in aspect-based sentiment analysis can potentially help in improving recommendation systems that depend on user reviews.

2.3 Spatio-temporal Factors for Recommendation

It is observed that users' interests tend to drift as they move across regions. Therefore, the spatio-temporal factors have been exploited for POI recommendation in many recent approaches [35, 42, 53, 55].

Yin et al. in Reference [53] introduce a Spatial-Temporal LDA (ST-LDA) technique for POI recommendation. A joint probabilistic generative model has been used in ST-LDA to understand region-based user interests. The model gives high importance to geographic influence by exploiting the content of POIs. Zhang et al. in Reference [55] perform a region-based POI recommendation and term the approach as cross-region collaborative filtering recommender system (CRCF). This work mainly focuses on data sparsity problem and the "new city" problem. Recommending POIs to a user who visits a region for the first time is described as the "new city" problem. The proposed approach utilizes the long-term and short-term preferences of a user. The top-K locations with similar long-term and short-term preferences are finally selected for recommendations. CRCF fails to recommend a cold-start POI within a "new city" to a user. The POI feature learning approach used here does not explore the sentiments from user reviews.

In Reference [42], a text-driven latent factor model (TLFM) has been proposed. It captures review semantics, user preferences, and product characteristics. Subsequently, they jointly model the user and product latent factor models. A pairwise rating comparison technique has been introduced for addressing the new user and unpopular POI recommendations. In this regard, the AvgUI method utilizes the average ratings by users and at POIs. Whereas, the UnkUI method randomly selects sample reviews for training. However, the geographical influence is not considered during recommendation and a completely new POI in a city cannot be recommended by the TLFM approach. Qian et al. in Reference [35] deal with another related recommendation problem termed as the cold-start spatiotemporal context. They proposed a spatiotemporal context-aware and translation-based recommendation framework (STA) for joint modelling of check-in time and location.

2.4 Neural Network–based Approaches for Recommendation

Very recently neural networks are also used for location recommendation. Kong et al. in Reference [23] predict locations to be visited by a user after a specified time window such as 30 minutes, 1 hour, 2 hours, and so on. An LSTM-based model is proposed in this approach, which consists of two modules. The first module addresses the data sparsity problem by combining the spatio-temporal influence of a location into LSTM (ST-LSTM). The second module uses an encoder-decoder network to incorporate the sequence of locations visited in a day into a hierarchical extension of ST-LSTM and term it as HST-LSTM. This work primarily focuses on predicting locations on an hourly basis, whereas our work presents a recommender system for notifying POIs to users based on their preferences. In Reference [7], a multi-view recurrent neural network model

(MV-RNN) is proposed to make sequential location recommendation. This neural network–based approach performs location recommendations, but uses images and other visual characteristics for learning POI features.

2.5 Addressing Cold-start Recommendation Scenarios

POI recommendations are evaluated in cold-start scenarios in References [10, 11, 12, 15, 26, 27, 33, 47, 48, 56]. The iGSLR framework [56] uses a kernel density function (KDE) [40] to compute rating of an active user to a new location. The distribution of visited locations, ratings, and social influences are incorporated into the KDE model. However, influence of nearest neighbors on a target user is not considered in the approach. Zhang et al. [12] proposed a context-aware approach by correlating the social network information and geographical distance on the social network. This approach addresses the user cold-start problem. However, getting sufficient context information is difficult.

User cold-start problem for location recommendation is addressed in Reference [11] by correlating social ties and geographical distances between target user and the social ties. Four types of friends, namely, local friends, local non-friends, distant friends, and distant non-friends are identified. The set of locations visited by these friends are ultimately considered for recommendation to the target user. Although the approach utilizes social ties as one of the influencing factors in recommendation, POI features from reviews are not explored in this approach. An implicit feedback-based content aware collaborative filtering (ICCF) framework is proposed in Reference [27]. As a user generally has accounts in various other social networks, the basic user profiles are explored from different social networks to tackle the cold-start user problem. The latent profile of cold-start users is estimated in Reference [48] by using the profiles of the warm users or the users having enough reviews/ratings. When a cold-start user rates a POI, then the available rating is compared with the ratings provided by the warm users at similar POIs and thus creates the user profile for cold-start users. The approaches in References [10, 11, 12, 27, 48] focus on the user cold-start problem and not specifically on the POI cold-start problem.

A generic graph-based embedding model (GE) is proposed by Xie et al. in Reference [47]. Four important factors, namely, geographical influence, sequential effect, temporal effect, and semantic effect, are combined together, and it is represented into a low-dimensional space. For cold-start POIs only the geographical influence and the semantic effects of available reviews are utilized. To capture the time varying user interests, a time-decay approach is utilized in the GE model. The recommendation approach in Reference [26] explores check-ins of friends. The framework involves two steps. First, the unvisited locations that are interesting to a user are identified based on preferences of friends. Second, check-in data of users and their corresponding friends are provided to a matrix factorization model to learn preferences. For standard recommendation the user preferences are fused with user distance from the concerned locations. The new locations are recommended by exploiting the neighboring locations. However, the approach fails to detect the newly evolved POIs in a region. Moreover, the approach does not consider the possibility to exploit cross-domain knowledge for gathering information on a new location.

The existing approaches look into the problem of POI recommendation from a user's perspective by learning user interests and socio-geographic influences. This has been proven to work for POI recommendations and addressing user cold-start problems. However, the POI cold-start or new POI problem is not explored in literature. A recommender system should have the knowledge of new POIs (CSPs). In this direction, Mazumdar et al. in Reference [33] proposed a crowdsourcing-based approach for addressing the cold-start POI problem (CPC). The CPC approach crowdsources multiple social networks for gathering information on the cold-start POIs. User preference over features of POIs are learned using sentiment analysis technique and, finally, top-K POIs are recommended based on the learned preference of the target. It can be noted here that our work is mostly influenced by Reference [33]. However, there exist multiple limitations of the prior work. In Reference [33], all the POIs in a region are considered for finding the top-K recommendations, which is not efficient. POIs usually have a wide range of features, and hence rating of them also has a high deviation. Therefore, all the features present in a POI may not be used for preference matching with the target user. We extend the work in Reference [33] by incorporating three new modules and follow a more suitable approach for addressing the cold-start POIs problem.

In the following section, limitations of existing approaches and the basic formulation of our proposed approach are described.

3 THE MOTIVATION

A location-based social network (LBSN) essentially comprises users, business enterprises or POIs, reviews, ratings, friends, and so on. Relationships among entities (users and locations) of an LBSN can be combined to develop an efficient recommender system. The contents (reviews and ratings) posted by users at various POIs, the social influence, and the geographical influence are the important factors for recommendation. These factors are used to compute the chances of an active user u to visit a new location l. The rating at l given by a user u can be estimated by exploring the ratings given by u at locations similar to l. In this case, similarity between locations is computed from their historical rating data. Social influence depends upon the ratings at a POI by an active user's friends or nearest neighbors. It can be noted here that all the above approaches primarily require the historical rating and review data at a location to consider it for recommendation.

A cold-start POI does not have historical rating or review data. Therefore, factorization and topic modelling–based approaches cannot be applied. It is hard to get the social influence, as a CSP does not have ratings from friends. Thus, the traditional approaches cannot be used directly to handle CSPs. Moreover, adopting the same strategy for cold-start user recommendation and cold-start POI recommendation may not be accurate. This is because of the fact that for both the approaches the intra-network data is generally used. Although it is possible to estimate cold-start user preferences from the intra-network warm user profiles, but estimating the dominating features of a cold-start POI having no historical data, from the available intra-network POIs is not meticulous.

Next, we discuss the limitations of state-of-the-art recommender systems in handling cold-start POIs.

(1) USG [52] combines user, social, and geographical influences during recommendations. The attributes or features of a POI that are of primary importance for a target user can only be understood from the review data. However, the work does not explore the reviews provided by target users at a POI.
(2) CRCF [55] approach selects the frequently used words in a review as the features. However, frequency alone cannot depict the sentiment associated with the word or feature. Moreover, this approach also fails to recommend a cold-start POI to a user who visits a "new city." This is due to the fact that the proposed algorithm has no scope of finding cold-start POIs and updating them in the system. Moreover, in a "new city" the POIs having features related to the user's precomputed preferences are only recommended. Therefore, a cold-start POI with no prior knowledge of prominent features is never considered for recommendation.
(3) Two POIs may belong to same category, but their prominent features may vary. For example, features and ratings of two venues A and B are represented as {cleanliness: 4, parking: 1} and {cleanliness: 1, parking: 4}, respectively. They both belong to restaurant category. According to the approach of venue similarity in SELR [49], a user u who

predominantly prefers cleanliness may tend to get recommendations of both POIs A and B even if venue B has poor rating for cleanliness. A recommender system should provide POIs that match the preferred features or preferences of a user. On a similar note, learning positive and negative features of a POI is also an important factor that should be considered before recommending POIs. SELR does not explore the implicit features of the POIs. Moreover, it fails to consider the cold-start POIs while developing the user-venue matrix. This restricts the proposed recommender system to provide cold-start POIs for a recommendation to a target user. From a business model perspective, a POI having dense historical data can only be interested for such a recommender system. Moreover, geographical influence is not considered in SELR. A POI near to the active user does not get any preference in the recommended list.

Motivation of the current work is obtained from the fact that cross-domain knowledge can be exploited for collecting related data on a cold-start POI. Moreover, extracting attributes or features of a POI and subsequently marking them as good or bad is a challenging task. Detailed description of the proposed feature-based POI recommendation approach of dealing with the cold-start POIs and recommending POIs to users is presented in the following section.

4 PROPOSED APPROACH

In this section, a detailed description of the proposed POI recommendation approach is depicted. It has five steps as follow:

(1) Inherent features and their corresponding ratings at a POI are identified from the available reviews. (*Identifying POI features*)
(2) A fuzzy clustering approach is used to group similar POIs on the basis of their corresponding features. (*Feature-based clustering of POIs*)
(3) The cold-start POIs are identified by crowdsourcing and subsequently their inherent features are extracted. (*Cold-start POI features*)
(4) The cold-start POIs are assigned to the clusters on the basis of their estimated ratings. (*Assigning cluster membership to cold-start POIs*)
(5) Traditional item-based collaborative filtering is applied to provide a list of POIs, and they are ranked on the basis of geographical distance for recommendations. (*Recommending POIs to users*)

4.1 Identifying POI Features

A point-of-interest possesses a certain set of features that attributes to its popularity or in some cases infamy. Therefore, identifying features for various POIs are studied here as the first step towards the POI recommendation. Reviews can be analyzed for understanding features of a POI and the corresponding rating illustrates their degree of likeness. It has been found that people tend to focus on certain features while providing a review [49].

The ratings provided by a user over a POI is either explicitly mentioned for a feature or implicitly for the POI as a whole. Advantages of an explicit rating is that we can easily understand the features of a POI, and naturally their likeness. The POI features obtained from such type of ratings are termed as the explicit features for a POI (*EF*). However, it has been observed that people often tend not to provide explicit rating at POIs to save time and sometimes reluctance to browse through the set of questionnaire for explicit ratings. Features that are not explicitly mentioned in the user ratings are termed as implicit features (*IF*) in the rest of this article. A very simple aspect-based POI implicit feature extraction technique (*APIF*) is used in this work.

Sentiment refers to a feeling or attitude towards an event or topic. Finding sentiments from a given statement can be formally defined as a quadruple (s, g, h, t), where s is the sentiment, g is the subject/target object on which the sentiment is conveyed, h denotes the person who provides the statement, and t is the time when the sentiment is expressed. The task of sentiment analysis can be broadly categorized into three types based on the target object: document-level, sentence-level, and aspect-level sentiment analysis [28]. The sentiment conveyed through the corpus of all reviews on a POI or product is computed at document level. In a sentence-level sentiment analysis, the target entities are the individual sentences that constitute a review. The document- and sentence-level analysis are used when we focus on a single entity. However, in a review generally the customer emphasizes various aspects or attributes of a single product or in the current context a POI. The task of sentiment analysis can be illustrated by an example. Let a review on a KFC restaurant in New York be, "I liked the food over here. The ambiance is soothing but the service is worse." The document-level sentiment analysis would identify this review on KFC as positive. A sentence-level analysis would identify the first sentence as positive and second as neutral (due to adjectives "soothing" and "worse"). Whereas, an aspect-based sentiment analysis would identify the aspects and their corresponding sentiments such as "food," "ambiance," and "service" having sentiments positive, positive, and negative, respectively. The proposed APIF technique identifies the aspects from a POI review and considers these aspects as the implicit features.

Let $U = \{u_1, u_2, \ldots, u_n\}$ be the set of all users, $P = \{p_1, p_2, \ldots, p_m\}$ be the set of all POIs, and $R = (r_1^1, r_1^2, \ldots, r_i^j)$ be all the reviews available. A review r_i^j denotes the review given by a user u_j at a POI p_i. Let U_{p_1} be the set of users who have reviewed at a POI p_1. Set of all available reviews at the POI p_1 can thus be denoted as

$$R_{p_1} = \bigcup_{x \in U_{p_1}} r_{p_1}^x. \tag{1}$$

A series of pre-processing techniques such as stop words removal, word tokenization, and so on, [45] are performed over the reviews. Further, POS tagging is performed using the Stanford CoreNLP library [32]. It has been observed that a review is mostly focused on the common nouns and compound nouns present in it [36]. Whereas, the proper nouns are not considered as aspects or features, as they are more specific to subjects that do not reflect a feature. Here, we select the combination of common and compound nouns as the aspects or features of a POI p_1. These features are implicit, thus, we need to identify their probable ratings. We analyze the sentiments expressed on the extracted features to estimate their ratings. Previous studies on NLP have found that sentiments on an aspect are expressed mostly through adjectives [16, 20, 36, 41]. In accordance to this finding, we associate all the adjectives related to each extracted aspect terms for every POI. We term the adjectives as sentiments and nouns as the aspect terms or implicit features (IF). For example, a review on a stadium, "the outfield is fantastic at Lords," has "outfield" as the common noun, which is interpreted as the aspect, and its corresponding sentiment is expressed with the adjective "fantastic." A sentiment distance threshold δs is used in this regard. A sentiment is associated with a feature if it remains within δs words of the feature and both the feature and the sentiment belong to same sentence.

The features are associated with multiple number of sentiments. Each sentiment has a polarity towards the associated feature. The SentiWordNet [1, 9] provides a corpus of possible sentiments and its polarity to identify whether it is used in positive or negative sense. In this work, the SentiWordNet polarities are used by calibrating them within a scale of 1 to 5. There are many sentiment words like "long" that can be used for both positive and negative sentiment. Therefore, in SentiWordNet, we get both values of positive and negative polarity. It can be noted here that in many cases, we find negative wordings in a review, which tends to reverse the sentiment expressed. For example, a review on a museum, "parking is not as big as I thought," expresses a negative association of sentiment "big" with the feature "parking." In APIF, we move (δn) word forward and backward directions with respect to a sentiment (here, *big*) to find the associated feature in a sentence. Thus, if a negative word is found within δn hop words of a sentiment, then we reverse its polarity. Inferred rating *IR* of a feature f for a POI p across all reviews on it is thus computed as,

$$IR_f^p = \frac{\sum_{i \in S_f} polarity(i)}{\left|S_f\right|}, \tag{2}$$

where
S_f = sentiments associated with feature f.

It can be noted here that the explicit ratings generally ranges between 1 to 5. Therefore, the inferred ratings for the implicit features obtained using *APIF* are also normalized in the same range between 1 to 5.

A POI-feature rating matrix M is developed with P number of rows presenting POIs and columns depicting all possible features Q. Each entry in M is the aggregated rating r_{pq} by all users on a feature q at a POI p. This matrix M is sparse, as a POI does not possess all features. In literature there exist many traditional techniques of missing value identification [13, 25, 31], incremental learning using variants of matrix factorization [24, 29, 38], and so on. However, these techniques may not work in this context. This is due to the fact that a feature like "outfield" can only be related to a stadium, park, or even a forest, but it can never be related to a restaurant, museum, or theatre. Therefore, in this work none of the above-mentioned techniques are used. For each POI in M, the features having high rating are treated as the representative or prominent features. On a similar note, the substandard features of a POI can also be learned from M. Mostly the features that are rated with "1" are considered as infamous. Detailed description of the APIF algorithm is depicted in Algorithm 1.

Knowledge of the good and substandard features of POIs can be extended in grouping them together. We need to employ a clustering technique to group the POIs on the basis of their features. The task of clustering the POIs on the basis of their positive or negative endowment of a feature is described in the following section.

4.2 Feature-based Clustering of POIs

The POI-feature rating matrix M described in the previous section can be utilized to perform clustering of POIs. Clustering on the basis of POI features can categorically organize similar POIs. In this regard, a feature can be good or bad with respect to POIs. Features such as cleanliness, price, amenity, and so on, can be marked as good or bad for a particular POI from its corresponding rating at a POI. Moreover, the total set of features does not belong to all the available POIs. However, the POIs do have very high overlapping set of features. For example, parking is a common feature for restaurant, stadium, and also for a temple. Similarly, the feature "ambience" is related to {restaurant, club} and feature "audio" more relevant to {auditorium, theatre}, and so on. Therefore, in context of this work, we require a clustering approach that can utilize the rating on features as the distance function and can place a particular POI in multiple clusters. In this regard, we adopt the standard fuzzy C-means clustering technique [3] in our problem. A toy example is provided to depict the suitability of fuzzy-based clustering technique in this work. A POI-feature rating matrix TM is presented in Table 1(a), having five POIs $TP = \{P_1, P_2, P_3, P_4, P_5\}$ and four features $TQ = \{F_1, F_2, F_3, F_4\}$. Each entry in TM is the inferred rating IR obtained from the APIF algorithm. Fuzzy C-means clustering is performed over matrix TM with data points as the POIs. Membership of each POI to the clusters is represented in Table 1(b).

ALGORITHM 1: Aspect-based POI implicit feature extraction (*APIF*)

Input: $\mathcal{D}$: Reviews at a POI p by all users
Result: *IF*: Implicit features of POI p
IR: Inferred rating of the implicit features
Data:
δs: sentiment distance threshold
δn: negative word distance threshold
S: selected sentiments for a feature
w_e: weight of an edge e connecting a sentiment node with a feature node
FSP: matrix storing feature, sentiment, and corresponding polarities
for *each sentence st in* $\mathcal{D}$ **do**
    $IF \leftarrow$ select all common and compound nouns from st;
    $Sentiments \leftarrow$ select all adjectives from st;
    for *each feature* f *in IF* **do**
        compute distance of each element in $Sentiments$ from f;
        $S \leftarrow Sentiments$ within δs distance from f;
        compute polarity of S from SentiWordNet;
        if *there exists a negative word within* δn *distance of sentiment* S **then**
            reverse the polarity value of the sentiment S;
        end
        store feature f, sentiment S, and corresponding polarity in *FSP*;
    end
end
compute *IR* of each feature in *FSP* using G;
return *IF* and *IR* ;

The membership value of a POI in this work signifies how much the POI is associated with a cluster. More specifically, a POI with low membership value to a cluster depicts that its inherent features are different from the other POIs having higher membership value in the same cluster. It can be noted here that our primary goal in clustering the POIs is to group similar set of locations having overlapping features and also learn the dominating features of the member POIs. In this regard, a POI with low membership value will hinder the task of selecting the dominating feature of a cluster. Therefore, we set a membership threshold of δc on the basis of which the cluster membership of a POI is decided. A POI is assigned to a cluster if it possesses a membership value of more than the threshold. For example, from Table 1(b) the membership of POI P5 is 0.03 and 0.97 to cluster 1 and 2, respectively. We assign the POI P5 to cluster 2 and not to cluster 1, as it does not satisfy the predefined cluster membership threshold. Similarly, the POI P4 is assigned to both the clusters (C1 and C2), as its membership value exceeds the threshold (Table 1(c, d)). The final clusters thus formed along with the data points are shown in Table 1(c) and 1(d). It can be observed that the feature-based ratings at POIs within cluster-1 signifies that its members have a very positive feedback for feature F2. Similarly, cluster-2 forecasts a negative opinion for features F3 and F4 at the member POIs. The features F2, F3, and F4 for POI P4 have almost similar rating as that of other member POIs in cluster-1. Similarly, ratings for features F3 and F4 at POI P4 is similar to the member POIs belonging to cluster-2. Therefore, it is sensible for POI P4 to be a member of both the clusters. Such findings inherently help a recommender system to understand the group of POIs having good and bad features. In continuation of this approach, two important questions arise. First, which top-F features from each cluster should be selected so they can represent majority of

Table 1. Example Showing the Feature-based Clustering of POIs Using Fuzzy C-means Clustering Technique

(a) POI-Feature rating matrix

	F1	F2	F3	F4
P1	5	4	1	1
P2	3	5	0	2
P3	4	4	1	0
P4	0	5	1	1
P5	1	0	1	1

(b) Membership values of POIs for each cluster after fuzzy clustering. (#clusters = 2 and #iteration = 15)

	C1	C2
P1	0.92	0.08
P2	0.92	0.08
P3	0.93	0.07
P4	0.59	0.41
P5	0.03	0.97

(c) Member POIs of Cluster-1 having membership threshold δc as 0.2.

	F1	F2	F3	F4
P1	5	4	1	1
P2	3	5	0	2
P3	4	4	1	0
P4	0	5	1	1

(d) Member POIs of Cluster-2 having membership threshold δc as 0.2.

	F1	F2	F3	F4
P4	0	5	1	1
P5	1	0	1	1

(e) Score *FS* for each feature in the clusters. The top-2 features in each cluster are marked in bold.

Cluster	F1	F2	F3	F4
C1	**0.71**	**2.85**	0.38	0.41
C2	0.15	0.28	**1**	**1**

(f) Dominating features for each cluster along with their *Impact* on the corresponding clusters.

C1	
F1	**F2**
4	4.5

C2	
F3	**F4**
1	1

(g) Feature vector for each cluster along with the corresponding feature impact is presented here. The impact of a non-dominating feature with FS > 0 is set as 1 to represent a bad sentiment of the feature in the cluster. Similarly, the impact of a non-dominating feature of a cluster with FS=0 is set as 0 to represent absence of the feature among the member POIs in the cluster. The top-2 features (marked in bold) are always selected as the dominating feature for each cluster.

Cluster	F1	F2	F3	F4
C1	**4**	**4.5**	1	1
C2	1	1	**1**	**1**

the member POIs? Second, does the selected feature portray a positive or negative opinion on the member POIs? To address these issues a representative feature selection technique is introduced.

To select representative features from a cluster, the *significance* and *popularity* of each feature are taken into consideration. The *significance* captures the coverage of a feature among member POIs and *popularity* considers how frequently a member POI received a rating on the feature. These two terms are used here to identify the top-F dominating features in a cluster. Formal definition of *significance* and *popularity* are given below.

Let $C = \{c_1, c_2, \ldots, c_d\}$ be the set of all clusters, $P = \{p_1, p_2, \ldots, p_m\}$ be the set of all POIs and $P^{c_1} = \{p_1, p_2, \ldots, p_n\}$ where, $n \leq m$, $P^{c_1} \subseteq P$, is the set of POIs in cluster c_1. Let TM^{c_1} be a POI-feature rating matrix for a cluster c_1 with n-rows as the member POIs P^{c_1} and q-columns as all the possible features $Q = \{f_1, f_2, \ldots, f_q\}$. An entry $IR^{c_1}_{p_1 f_1}$ denotes the rating at POI p_1 for feature f_1 in the cluster c_1.

Definition 1 (Significance). Significance of a feature f_1 ($f_1 \in Q$) in a cluster is the number of POIs in P that possess it as an attribute. It can be computed as:

$$Sig^{c_1}_{f_1} = \frac{count(IR^{c_1}_{*f_1} \geq 1)}{|P^{c_1}|}, \tag{3}$$

where $IR^{c_1}_{*f_1}$ is the ratings for feature f_1 in TM^{c_1} by all POIs in P^{c_1}.

The frequency at which a POI is visited by a user is not considered in *Significance*. To consider frequency as a factor, we define the *popularity* of a POI in a cluster.

Definition 2 (Popularity). Popularity of a feature f_1 ($f_1 \in Q$) in a cluster c_1 is the fraction of the number of users who have visited POIs having the feature f_1 in c_1. It can be computed as:

$$Pop^{c_1}_{f_1} = \frac{\#users\ visited\ POIs\ having\ a\ feature\ f_1\ in\ c_1}{\sum_{POI_i \in c_1} \#users\ visited\ POI_i}. \tag{4}$$

Another important factor for representative feature selection is that of the variation of the ratings for a feature. Features with less variation in obtained ratings are generally considered as the representative features. Thus, for finding dominating features, we compute the standard deviation σ of the ratings at each feature for identifying the representative features of a cluster. Therefore, the proposed technique of representative feature identification combines four factors: the rating of features at POIs, number of POIs associated with the features, number of users who have visited at the POIs that are associated with the features, and the standard deviation of the ratings for the features at all POIs. These four factors are combined together to compute a score FS for each feature in a cluster. Score of a feature f_1 in a cluster c_1 is computed as:

$$FS^{c_1}_{f_1} = \frac{Sig^{c_1}_{f_1} \times Pop^{c_1}_{f_1}}{1 + \sigma^{c_1}_{f_1}}. \tag{5}$$

Top-F features from each cluster having highest FS are selected as the dominating features. In the toy example, for simplicity, we set a condition that the rating at a POI is performed by one user and none of the users have rated more than one POI. The features (F1, F2) and (F3, F4) are found to have high FS in cluster C1 and C2, respectively, when top-two cluster features are selected (Table 1(e)). The selected top-F dominating features in a cluster is referred to as the cluster features throughout the rest of this article. Next, there is a need to quantify the cluster features. In this regard, we introduce the term *impact*, which helps us to compute the average rating of the cluster features. The *impact* of a dominating feature f_1 in cluster c_1 can be defined as follows:

Definition 3 (Impact). Impact of a feature f_1 in a cluster c_1 is the average rating of f_1 at POIs in the cluster c_1.

$$Imp^{c_1}_{f_1} = \frac{\sum_{i \in P^{c_1}_{f_1}} IR^{c_1}_{i f_1}}{\left|P^{c_1}_{f_1}\right|}, \tag{6}$$

where
$P^{c_1}_{f_1}$ = POIs in cluster c_1 having f_1 as a feature
$IR^{c_1}_{i f_1}$ = rating for feature f_1 at POI i in cluster c_1.

Table 1(f) shows the cluster features and their *Impact* on the clusters for the toy example. The *impact* of a feature on a cluster that is not selected as a dominating feature is considered as 1 if its FS is greater than 0, otherwise the *Impact* is set as 0 for $FS = 0$ (Table 1(g)). This is due to the fact that, in the proposed approach, we use 0 to signify that a feature is not present in a POI and 1 to signify a bad sentiment towards a feature or a POI. Therefore, to represent a low *impact* of the non-dominating features, we set them as 1 in the cluster feature vector. For each cluster, the process of dominating feature extraction and computing their *impact* is repeated every time a new POI becomes its member. This process can be periodically updated using a temporal delay of, say, one month or even earlier as per requirement. Next, we need to identify the cold-start POIs for including them in the generated clusters. In the following section, we focus on finding the cold-start POIs and, subsequently, learn their associated features.

4.3 Learning Cold-start POIs and Their Features

Learning features of a cold-start POI is a challenging task due to the absence of historical review and rating data from users. Any POI (say, p) that is not enlisted in a social network (say, s) is considered as "new" with respect to that particular online platform. However, POI p may have been enlisted in other social networks. For finding cold-start POIs for any social network s, we crawl other available online platforms that enlist POIs in a region (say, r). The proposed recommender system is based on the Yelp social network. Therefore, a POI that is not present in the Yelp database is considered as a CSP with respect to Yelp. Our approach towards collecting information about cold-start POIs depends upon a fusion of various online social media. Generally, POIs provide descriptions about their business through social media platforms such as Foursquare, Google places, Facebook, Twitter, and even official websites. In this work, we explore Foursquare and Google places API. The data (reviews and ratings) on "new" POIs are provided into Foursquare and Google places by their real-life users who either reside in the region or have visited the locations in the past. Therefore, we depict this process of finding cold-start POIs from multiple online social networks as crowdsourcing.

Foursquare is a popular location-based social network that provides a publicly available API that can be crawled to collect online information. The Foursquare API provides a POI searching facility through its *venues* endpoint. Information such as venue name, address, number of check-ins, reviews, distance of a venue from the query location, and so on, can be obtained from it. Similarly, the Google places API provides the *nearbysearch* endpoint for gathering information on various places of interest. However, finding CSPs only by crawling publicly available APIs has certain issues. Location of existing POIs residing in the same city that are δd distant from each other are selected as the samples. Here, the distance threshold δd is set on the basis of the restrictions provided by the API web services used. For example, Foursquare venue search endpoint supports maximum 100 kms radius, thus it can provide details of all POIs residing within this distance from the sample location. Similarly, the *nearbysearch* request of Google places API web service supports maximum 50 kms radius. In this work, the maximum possible distance provided by the respective APIs is considered as the distance threshold δd. Set of all POIs obtained from the response of *search* (Foursquare) and *nearbysearch* (Google) endpoints are stored. The identifiers of the cold-start POIs are used in *tips* (Foursquare) and *details* (Google) endpoints to obtain the related reviews from social media. Further, this cross-domain knowledge on a cold-start POI is exploited for addressing the "new" POI problem for Yelp social network. It must be mentioned here that Foursquare has stopped providing ratings for the *tips* endpoint, and Google places provides only implicit rating at a POI for the *details* endpoint. Therefore, we apply the APIF algorithm (Section 4.1) for identifying the inherent features for each CSP and infer ratings on them. Detailed description of the proposed technique to identify the cold-start POIs and crowdsourcing the review and rating data on them is depicted in Algorithm 2.

ALGORITHM 2: Cold-start POI identification and social media data retrieval (CPISDR)

Input: $\mathcal{D}_C$: POI dataset for any confined geographical region C in Yelp
Result: CSP_C: Cold-start POIs in C
R^C_{CSP}: Reviews at the cold-start POIs CSP_C in C
Data:
$samples_C$: sample locations in C used for finding cold-start POIs
$fpid$, $gpid$: POI identifiers from Foursquare and Google Places, respectively
fpn, gpn: POI name from Foursquare and Google Places, respectively
fpa, gpa: POI address from Foursquare and Google Places, respectively
fll, gll: POI location co-ordinates from Foursquare and Google Places, respectively
ft, gt: time of review at a CSP from Foursquare and Google Places, respectively
frv, grv: review at a CSP from Foursquare and Google Places, respectively
δd: distance threshold (100 kms for Foursquare and 50 kms for Google Places)
L = location co-ordinates of all POIs in C;
$samples_C$ = first location in L;
for *each location l in L except the last location* **do**
    if *distance($l, l+1$) $\geq \delta d$* **then**
        $samples_C$ = store location ($l+1$) as a sample location in C;
    end
end
for *each location l in $samples_C$* **do**
    [$fpid$, fpn, fpa, fll] = venues_search(l, δd) ; // Foursquare
    [$gpid$, gpn, gpa, gll] = place_nearbysearch(l, δd) ; // Google Places
    if *any record in [$fpid$, fpn, fpa, fll] or [$gpid$, gpn, gpa, gll] does not exist in $\mathcal{D}_C$* **then**
        CSP_C = store the new cold-start POI $fpid$ or $gpid$;
        [ft, frv] = venues_tips($fpid$) ; // Foursquare
        R_{fpid} = store ft and frv at POI $fpid$;
        [gt, grv] = place_details($gpid$) ; // Google Places
        R_{gpid} = store gt and grv at POI $gpid$;
        $R^C_{CSP} = R_{fpid} \cup R_{gpid}$;
    end
end
return CSP_C and R^C_{CSP};

To this end, the cold-start POIs are detected, their implicit features are identified, and corresponding ratings are obtained by crowdsourcing. Next, it is necessary to assign the newly found cold-start POIs, a membership to the generated clusters in Section 4.2.

4.4 Assigning Cluster Membership to Cold-start POIs

In Section 4.2, a set of clusters is formed where each consists of non cold-start POIs that underline the dominating features of the respective clusters. In a similar line, a new CSP also needs to be a member of clusters that represents equivalent features. We compute similarity between the inherent features of CSPs and the top-F dominating features for each cluster. Here the similarity is computed on the basis of the inferred rating IR at the inherent features of CSPs and the feature vector for each cluster. Let $C = \{c_1, c_2, \ldots, c_d\}$ be the set of clusters and $A^{c_i} = \{a_1, a_2, \ldots, a_F\}$ be the feature vector for a cluster c_i. Let CP be a cold-start POI having implicit features $CPF = \{cf_1, cf_2, \ldots, cf_t\}$ with corresponding inferred ratings $CPIR = \{cr_1, cr_2, \ldots, cr_t\}$. The cosine similarity measure [37] is utilized to compute similarity between the feature vector A^{c_i} of cluster c_i

and the inferred feature rating vector *CPIR* of *CP*. If any feature f_i in *A* does not exist in *CPF*, then the implicit rating of f_i in *CPIR* is considered as 0, and conversely the *impact* in *A* is also treated as 0 for a feature that is present in *CPF* but not in *A*. Likewise, we compute similarity between *CP* and feature vectors of all clusters formed from Section 4.2. The top-*CC* clusters having highest similarity score with *CP* are selected and *CP* is assigned membership of all of them. This allows a cold-start POI to belong with other POIs carrying similar features. There can be a scenario when no crowdsourced data are available for a CSP. To address such a scenario, we identify the geographically nearest POI with respect to the CSP. The top-*CC* clusters having the majority of the identified POIs are selected and the CSP is assigned to them. This completes the task of grouping existing POIs and cold-start POIs on the basis of features. As already mentioned in Section 4.2, the representative features and their associativity towards a cluster is re-computed as a cold-start POI is assigned its membership. Next, we recommend POIs to users by utilizing the knowledge gathered from the previous sections.

4.5 Recommending POIs to Users

This section first identifies the user preferences and then recommends POIs in accordance with the learned preferences. An active user generally has historical review and rating data at various POIs. It is plausible to learn a user's interest from the reviews provided at the visited POIs. By taking this into consideration, we utilize the APIF algorithm (Section 4.1) for learning user preferences. Set of all reviews published at various POIs by an active user is taken as input to the APIF algorithm. As an output, we obtain the features as preferences of the user along with the inferred ratings as those features that emphasize the degree of liking. Subsequently, we learn how the latent factors influence user ratings at various features. In this regard, a popularly accepted biased matrix factorization approach is utilized [24]. Here, we perform factorization on the user-feature rating matrix. Set of users for whom at least 5 features (preferences) can be extracted by using APIF algorithm are selected for factorization. Similarly, we use top-20 features with high rating as set of features in factorization. The parameters used for biased matrix factorization are mentioned in the experimental section. The latent features learned after factorization are subsequently used for personalized recommendation.

Forming clusters of POIs based on similar features (Section 4.2) and updating them with the CSPs (Section 4.4) help in efficiently computing the top-*K* recommendations. Instead of estimating the ratings at all the unvisited POIs for an active user, we compute a similarity score between the active user and all clusters. The similarity is computed on the basis of the feature vector for the active user and the feature vector of each cluster. The cluster with the highest similarity score is selected in the proposed approach. Evidently all the POIs that belong to the nearest cluster are chosen for recommendation to the target user. Two important issues arise here: first, the geographical factor of the POIs is not considered during recommendations; second, how to rank the POIs in a cluster to select the top-*K* recommendations. We propose a strategy that can address both the issues. A *RecScore* is computed for each of the member POIs of the cluster that have the highest feature-based similarity score with respect to the target user. The *RecScore* combines both the factors of spatial distance and chances of a user to visit a POI. An item-based collaborative filtering approach [39] is adopted here for computing the chances of a user to visit a POI in the nearest cluster. This approach considers the set of POIs previously rated by the target user and subsequently computes similarity of those POIs with the target POI in the nearest cluster. *RecScore* of a POI *p* in a cluster *c* with respect to a user *u* can be determined as:

$$RecScore^{u}_{pc} = \frac{1}{gdist(ql,p)} \left[\bar{r}_p + \frac{\sum_{z \in Z} sim(p,z) \times (r_{uz} - \bar{r}_z)}{\sum_{z \in Z} |sim(p,z)|} \right], \tag{7}$$

where
$gdist(ql, p)$ = geographic distance between the active user's current location ql and location of target POI p;
$\bar{r}_p$ = average rating at POI p in the nearest cluster;
Z = set of all POIs rated by user u in the cluster c;
r_{uz} = rating of user u at POI z;
$sim(p, z)$ = similarity between POIs p and z.

The rating scale of different users varies significantly. Therefore, to compute the similarity score between a pair of POIs, we apply the adjusted cosine similarity measure that is proved to be robust to such a scenario [39]. The adjusted cosine similarity between two POIs p and z can be computed as:

$$sim(p, z) = \frac{\sum_{u \in U} (R_{u,p} - \overline{R}_u)(R_{u,z} - \overline{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,p} - \overline{R}_u)^2}\sqrt{\sum_{u \in U} (R_{u,z} - \overline{R}_u)^2}}, \tag{8}$$

where
U = set of users having rating at both the POIs p and z;
$R_{u,p}$ = rating of user u at POI p;
R_u = average rating by any user u.

Generally, POIs with less number of ratings and reviews are rarely recommended due to their limited knowledge. However, in the proposed FPR approach, the details of cold-start POIs are collected by crowdsourcing. CSPs are given equal importance and are placed in clusters with POIs of similar characteristics. Even the inferred ratings of features at POIs do not depend upon the number of users who visited them, and this helps in considering cold-start POIs in the same line of the existing non-cold-start POIs. To evaluate the proposed POI recommendation approach, a series of experiments was performed that is described in the following section.

5 EXPERIMENTS

We performed several experiments for comparing the recommendation qualities of our proposed FPR approach with four existing state-of-the-art techniques USG [52], SELR [49], CRCF [55], and CPC [33].

5.1 Dataset Description

Experiments are performed using the publicly available Yelp dataset. Yelp shares large-scale real data for research purposes.[1] The Yelp dataset consists of 77,445 POIs of various cities in the U.S., Canada, UK, and Germany. We process the review dataset of all POIs at the granularity level of a state instead of a city; for example, all POIs belonging to cities such as Pittsburgh, Carnegie, Rankin, and so on, are grouped and considered as all POIs in Yelp within the state of Pennsylvania. Total number of states that registered a POI in the dataset are 27. Figure 1 shows the comparison of the number of recorded POIs at various states. It is observed that the top-eight states have more than 1K POIs, and they cover 98% of the total number of POIs in the Yelp dataset. The working dataset consists of the combined data of all the eight states: 75,762 POIs, 2,198,341 reviews, and 544,916 users.

5.2 Evaluation Technique

The evaluation strategies are devised in such a way that the proposed recommendation approach as a whole and its performance in handling CSPs can be tested. Experiments are first performed over

[1] https://www.yelp.com/dataset_challenge.

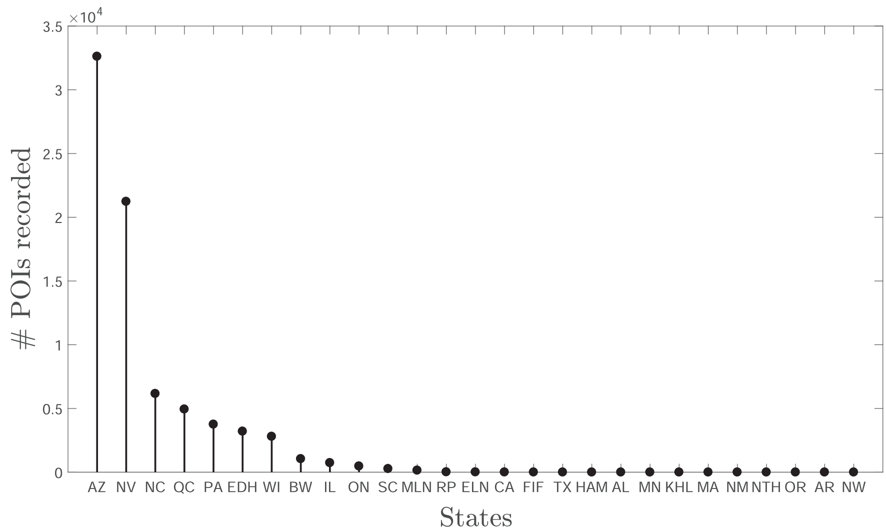
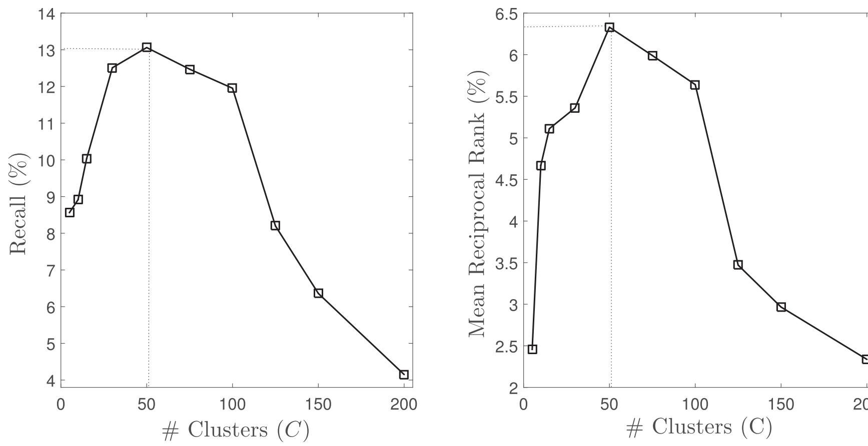


Fig. 1. Number of POIs recorded at various states in the Yelp dataset is presented here. The horizontal axis is marked with the acronyms of all the states (for example, AZ refers to Arizona, U.S.). The vertical axis shows the number of POIs in log scale. The states in the horizontal axis are arranged in descending order of the number of recorded POIs.

all the eight states individually and the results are recorded. We analyze performance of FPR and existing recommendation approaches for each of the eight states separately. Subsequently, all the eight states are combined as a single geographical location and the recommendation approaches are evaluated over it. Below, we mention the two strategies in detail.

(1) **Proposed POI Recommender**: We randomly select 10%, 20%, 40%, 60%, 80%, and 90% POIs from the historical check-in records of users to generate six training datasets. Rest of the data in each of the six cases are used for testing purpose (ground truth). This allowed us to check the variation in performance of the proposed system when less number of historical data is available. We evaluate the following two important aspects for each of the competitive POI recommendation approaches:
   i. Accuracy of the POI recommender.
   ii. Ranking of a relevant POI in the recommended list.
(2) **Handling CSPs**: For a recommender system, the task of handling CSPs involves identifying the newly evolved POIs in a city and then recommending them to the target users. In this regard, we first check the number of new POIs (CSPs) that can be identified for the selected eight states of the Yelp dataset by exploiting the contents of Foursquare and Google places. It can be noted here that these POIs are completely new to Yelp. Next task is to check whether the identified CSPs are recommended to the target users or not. In this regard, the following strategy has been used for evaluating accuracy of the proposed recommender system as follows: For each of the eight states, we arranged the member POIs in decreasing order of the number of ratings available on them. The bottom 10% of the POIs in each of the eight states are collectively considered and marked as the cold-start POIs (ground truth) for the Yelp dataset. Total 7,573 POIs were marked as the CSPs and the training dataset is generated by removing these marked CSPs. They are further considered for evaluating the accuracy in selecting and ranking the recommendations. A similar procedure is followed when we experimented on each of the



(a) 50% Training data, K=20

(b) 50% Training data, K=20

Fig. 2. Recall (a) and Mean reciprocal rank (MRR) (b) of our approach on varying the number of clusters (C) from 1 to 200. Vertical axis shows the obtained Recall/MRR, and horizontal axis depicts the number of clusters used. In both the figures, the best-performing setting for our approach is highlighted using dotted lines to show the number of clusters (C) and corresponding Recall/MRR values.

eight states individually. For checking performance of POI recommenders in handling CSPs, we evaluate on the four ways given below.

i. Collecting CSPs for Yelp dataset.
ii. Successful identification of the marked CSPs in Yelp.
iii. Accuracy in recommending the marked CSPs in Yelp.
iv. Ranking of the marked CSPs in the recommended lists.

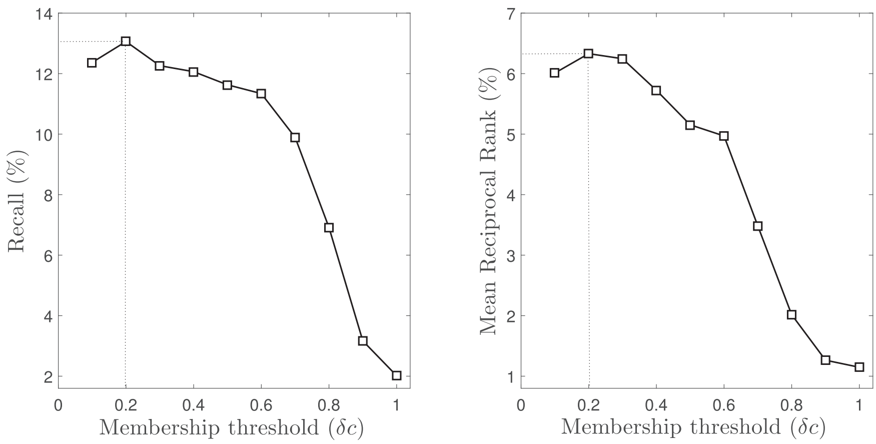
Our approach involves multiple number of parameters, and it is required to set a threshold for each of them. In the following section, we provide the description on the thresholds used and how they are selected.

5.3 Parameter Selection

Our approach involves few thresholds that can be listed as the sentiment distance threshold δs, hop words of a sentiment δn, number of fuzzy clusters C, membership threshold δc, number of clusters with CSPs (CC), regularization and learning rate parameters for matrix factorization. A series of experiments was performed to select the optimum value for all these parameters. Due to sparsity of the dataset, we cannot use the hold-out dataset for parameter setting. Therefore, for parameter-setting purpose, we use 50% data for training and the rest for testing.

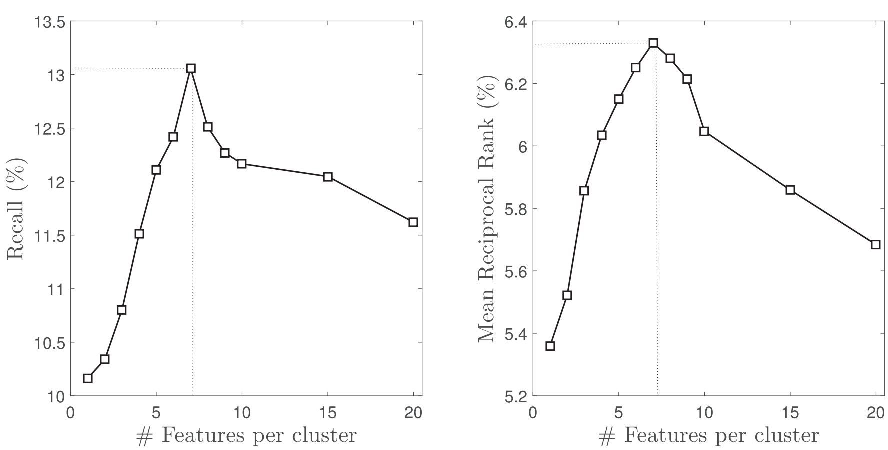
Figure 2 shows the Recall and Mean reciprocal rank (MRR) of our approach by varying the number of clusters from 1 to 200 on 50% training data and top-20 (K) recommendations. The performance of our approach was found to decrease gradually when the number of POIs was divided into more (above 50) number of clusters. At 50 number of clusters the presented method was found to perform best. Therefore, the number of clusters C for fuzzy clustering (Section 4.2) is set at 50 during the experimentation.

The membership threshold δc during fuzzy C-means clustering was increased gradually to the maximum value 1, and the performance was recorded (Figure 3). It was observed for both Recall and MRR that with increase of membership above 0.5 the performance is highly reduced. Therefore,



(a) 50% Training data, K=20 (b) 50% Training data, K=20

Fig. 3. Recall (a) and Mean reciprocal rank (MRR) (b) of our approach on varying the membership threshold (δc). Vertical axis shows the obtained Recall/MRR, and horizontal axis depicts the membership thresholds used. In both the figures, the best-performing setting for our approach is highlighted using dotted lines to show the membership threshold (δc) and corresponding Recall/MRR values.

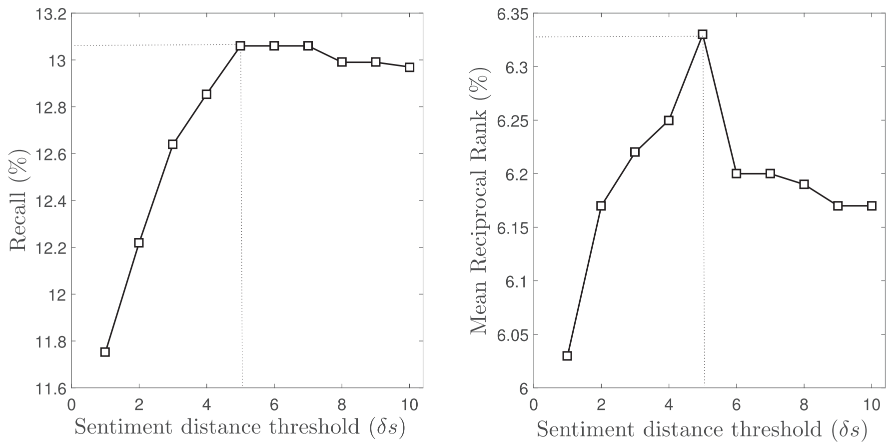


(a) 50% Training data, K=20 (b) 50% Training data, K=20

Fig. 4. Recall (a) and Mean reciprocal rank (MRR) (b) of our approach on varying the number of dominating features (F) from each cluster. Vertical axis shows the obtained Recall/MRR, and horizontal axis depicts the number of features used. In both the figures, the best-performing setting for our approach is highlighted using dotted lines to show the number of features (F) selected per cluster and corresponding Recall/MRR values.

we set the membership threshold δc at 0.2, which produces best performance for our method (Section 4.2).

The number of dominating features F from each cluster (Section 4.2) is another important parameter that is closely related to performance of our proposed recommendation systems. Figure 4 shows the results when selection of the number of dominating features is increased from 1 to 20. It was observed that selecting less number of features, i.e., the most prominent features of the grouped POIs, does not improve performance. Similarly, as we start selecting all features as



(a) 50% Training data, K=20 (b) 50% Training data, K=20

Fig. 5. Recall (a) and Mean reciprocal rank (MRR) (b) of our approach on varying the sentiment distance threshold (δs). Vertical axis shows the obtained Recall/MRR, and horizontal axis depicts the sentiment distance threshold used. In both the figures, the best-performing setting for our approach is highlighted using dotted lines to show the sentiment distance threshold (δs) and corresponding Recall/MRR values.

dominating, then the performance gradually decreases. A reason for such a behavior can be due to the fact that selecting more number of features from each cluster increases the feature-based similarity between the produced clusters. By selecting seven features per cluster as dominating, we recorded the best performance.

Generally, reviews are structured in such a way that the expressed sentiments (adjectives) are near to the subject (noun). A distance threshold is set on the number of words around an aspect that can be considered as the corresponding sentiment. In this regard, Figure 5 shows the performance of our approach with varying sentiment distance threshold (δs). Performance does not differ much after we set the threshold at more than 5. Whereas, as we increased the threshold from 1 to 3, significant improvement was observed. In experimentation, we set this threshold at 5, which is found to be optimal (Section 4.1).

Similar to the sentiment distance, the negative hop word distance (δn) is also tested by varying the threshold from 1 to 10 (Figure 6). After a certain point ($\delta n = 4$), we observe that the performance is not affected and the results are static. This is because we do not get enough negative words after threshold 4 that are related to the expressed sentiment (Section 4.1).

The number of clusters (*CC*) having highest similarity score with respect to a cold-start POI depends upon how much the recommender system needs to boost a new POI (CSP) for recommendation. Figure 7 shows how we select this threshold for our approach. The crowdsourced CSPs, even if assigned to more number of clusters based on similarity score, still do not affect our final result. It can be noted here that the recall is computed based on the number of CSPs that have been selected for recommendation. For the Yelp dataset, it is observed that even if we assign a CSP to more number of clusters, it is never selected for recommendation. This is due to the fact that by increasing number of clusters in which the CSP is assigned, we are actually assigning a CSP to dissimilar clusters (dissimilar with respect to the feature of CSP). Therefore, in this work, we set a threshold of 2 clusters for each CSP (Section 4.4).

For matrix factorization, we set the regularization parameter at 0.0001 and the constant for determining the rate of approaching minima at 0.01, as it is a standard approach [34, 55] (Section 4.5).

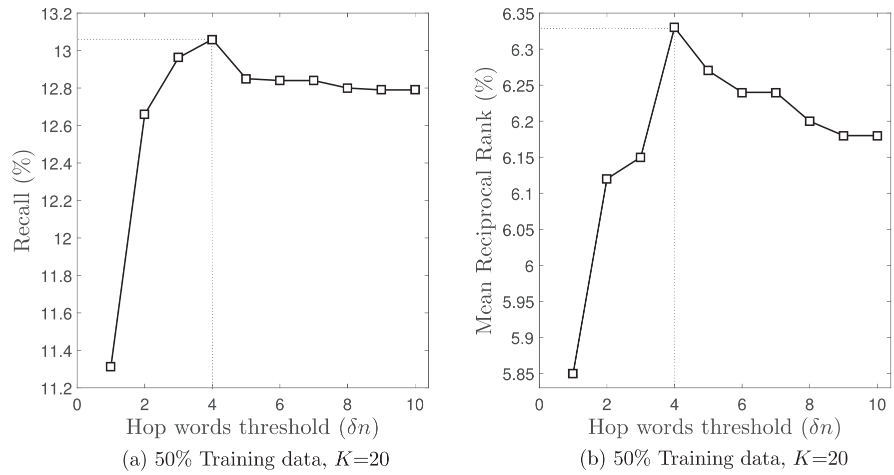


Fig. 6. Recall (a) and Mean reciprocal rank (MRR) (b) of our approach on varying the hop words threshold (δn). Vertical axis shows the obtained Recall/MRR, and horizontal axis depicts the hop words threshold used. In both the figures, the best-performing setting for our approach is highlighted using dotted lines to show the hop words threshold (δn) and corresponding Recall/MRR values.

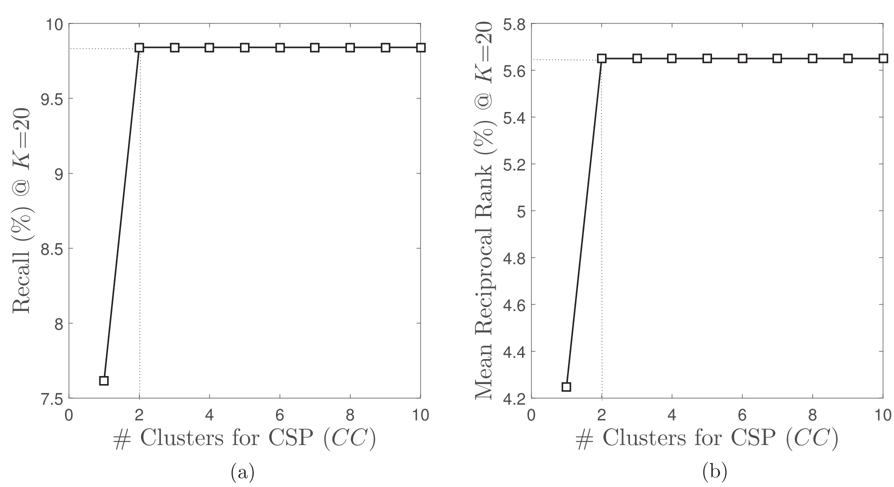
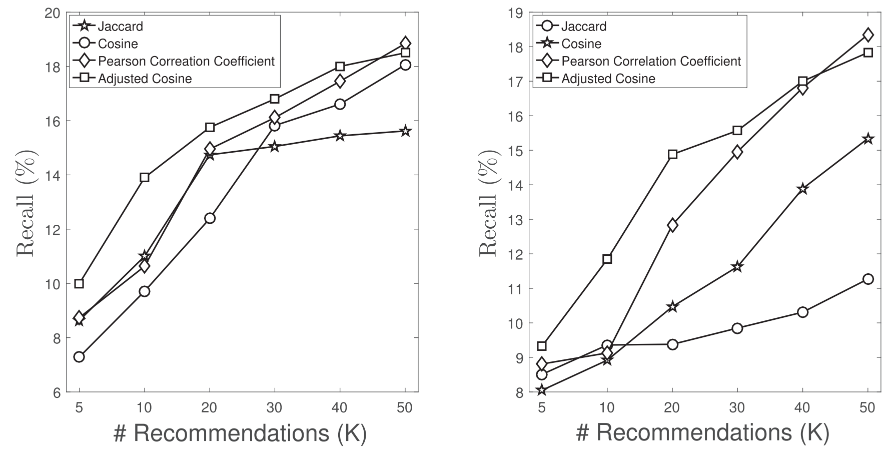


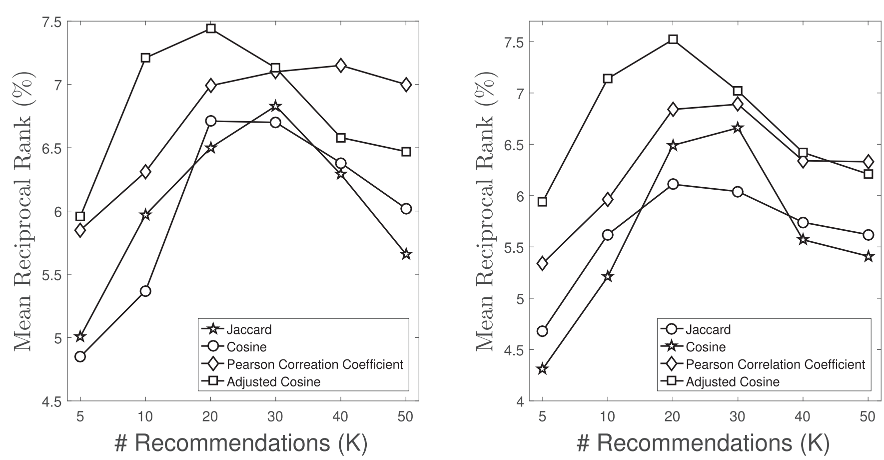
Fig. 7. Recall (a) and Mean reciprocal rank (MRR) (b) of our approach on varying the number of clusters (CC) assigned to a CSP. Vertical axis shows the obtained Recall/MRR, and horizontal axis depicts the number of clusters used for CSP assignment. In both the figures, the best-performing setting for our approach is highlighted using dotted lines to show the number of clusters for CSPs (CC) and corresponding Recall/MRR values.

In Section 4.5, we mentioned about using the adjusted cosine similarity for computing similarity scores between a pair of POIs. In this regard, other popular similarity metrics can also be used in place of the adjusted cosine similarity metric. Popular similarity metrics are Jaccard similarity, Cosine similarity, Pearson correlation coefficient, and so on. In the current work, we tested performance of our approach by using different similarity metrics. Figure 8 and Figure 9 show the Recall and MRR, respectively, for our approach by using different similarity metrics such as Jaccard, Cosine, Pearson correlation, and Adjusted cosine similarity. We tested for 90% and 80%



(a) 90% Training data (b) 80% Training data

Fig. 8. Recall of our approach by using various similarity metrics. Vertical axis shows the obtained Recall, and horizontal axis depicts the number of recommendations performed. Here, we show the results with 90% training (a) and 80% training data (b).



(a) 90% Training data (b) 80% Training data

Fig. 9. Mean reciprocal rank (MRR) of our approach by using various similarity metrics. Vertical axis shows the obtained MRR, and horizontal axis depicts the number of recommendations performed. Here, we show the results with 90% training (a) and 80% training data (b).

of training data with number of recommendations varying from 5 to 50. The Jaccard similarity is observed to be performing better for our approach compared to the cosine similarity measure for less number of recommendations. However, as we increase the number of recommendations, the Jaccard similarity has the lowest performance for our approach. For the present approach and experimentation in the Yelp dataset, we found that our proposed approach with adjacent cosine similarity measure was found to be performing better than all other measures for less number of recommendations. Pearson correlation coefficient was found to be the closest comparable metric for this particular work. However, our approach with Pearson correlation coefficient was found to perform better as we increase number of recommendations (K) above 40. A recommender system that can produce high accuracy with less number recommendations is believed to be better.

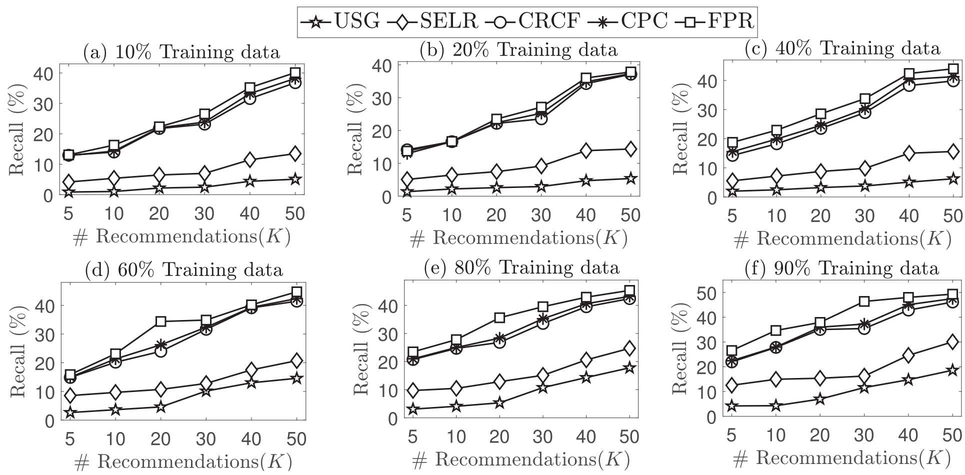


Fig. 10. Recall of the various POI recommendation approaches for the Arizona (AZ) state. The training set is varied from 10% to 90% for evaluating performance with sparse and dense data, respectively. The horizontal axis of each figure is the number of recommendations (K) that varies from 5 to 50. The vertical axis depicts the obtained recall in percentage.

Therefore, we utilize the adjacent cosine similarity metric in this proposed approach for computing similarity between a pair of POIs. The following section presents the obtained results with the above-mentioned parameter settings.

5.4 Results and Analysis

This section presents the results of the proposed recommender system and, subsequently, its ability to handle the cold-start POIs.

5.4.1 Proposed POI Recommender.

i. Accuracy of the POI recommender:

Accuracy of the recommendations is computed using the popular *recall* metric [55]. It is the ratio of number of recommended POIs that are relevant and the total number of POIs in the test dataset.

$$Recall = \frac{1}{|U|} \sum_{i \in U} \frac{\#Hits_i}{\#Ground\ truths_i}, \tag{9}$$

where
$Hits_i$ = number of recommended POIs that are relevant to user i;
$Ground\ truths_i$ = number of relevant POIs in the test dataset for user i.

We computed recall for each of the selected states separately. Arizona (AZ) and Nevada (NV) cover around 77% of the total records in the Yelp dataset. Wisconsin (WI) and Karlsruhe (BW) are the two states having lowest number of records in Yelp. They together contribute only 5% of the total dataset. Experimenting on all the states individually helped us to analyze performance of proposed FPR for both the sparse and dense datasets. Here, we report results of accuracy of the compared recommender systems for all the eight states individually (Figures 10–17) and subsequently, also provide results on the total dataset combining all the eight states together (Figure 18). The training set is varied from 90% to 10% for evaluating the recommender systems in sparse and dense dataset scenarios. In

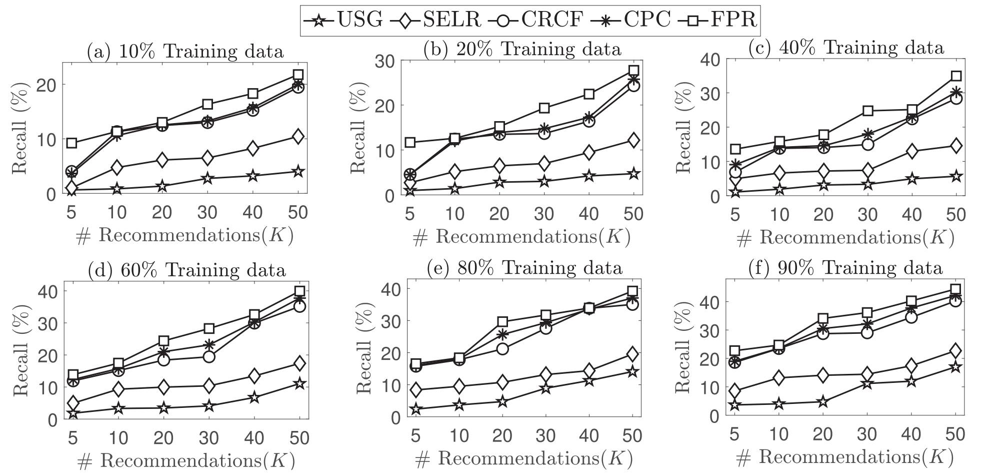


Fig. 11. Recall of the various POI recommendation approaches for the Nevada (NV) state. The training set is varied from 10% to 90% for evaluating performance with sparse and dense data, respectively. The horizontal axis of each figure is the number of recommendations (K) that varies from 5 to 50. The vertical axis depicts the obtained recall in percentage.

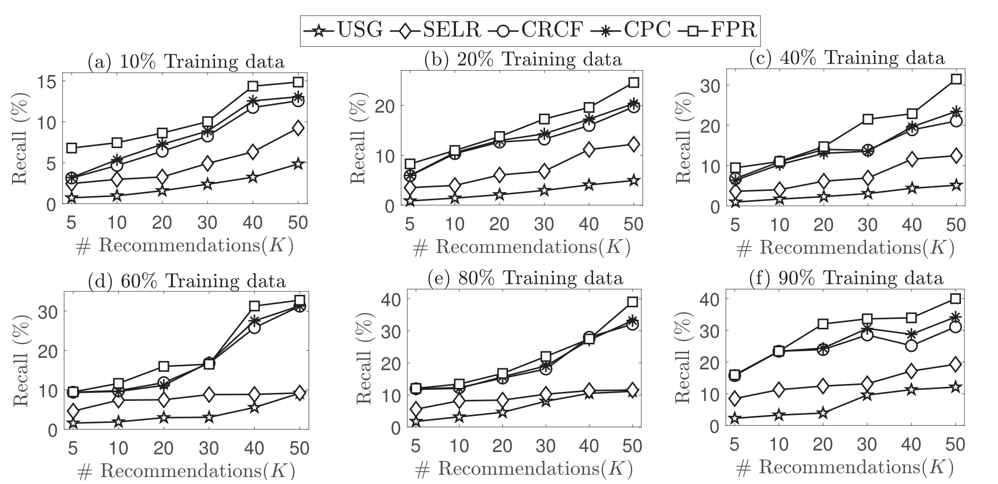


Fig. 12. Recall of the various POI recommendation approaches for the North Carolina (NC) state. The training set is varied from 10% to 90% for evaluating performance with sparse and dense data, respectively. The horizontal axis of each figure is the number of recommendations (K) that varies from 5 to 50. The vertical axis depicts the obtained recall in percentage.

addition to this, we also vary the number of recommendations (K) from 5 to 50. For Arizona, with 90% training data and 5 recommended locations, the proposed FPR approach produces accuracy of 26.55%, whereas the nearest approaches CPC and CRCF produce 22.4 % and 22% of accuracy, respectively. Similarly, for sparse data with 10% records used as training set, proposed FPR produces better accuracy than existing recommender systems (CPC, CRCF, SELR, and USG). In Karlsruhe (BW), with less number of records, we observe a major decrease in performance of all the compared recommendation approaches including our proposed FPR. The most accurate recommendation for FPR is obtained as 49.3% for

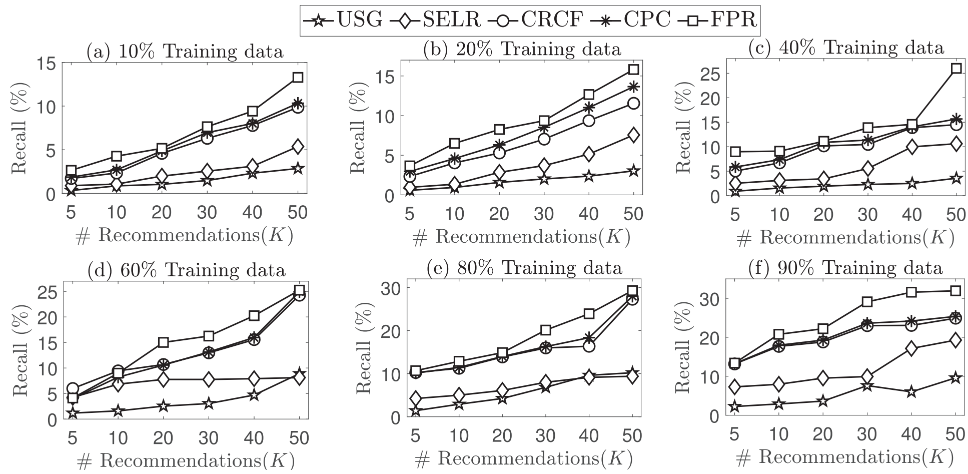


Fig. 13. Recall of the various POI recommendation approaches for the Quebec (QC) state. The training set is varied from 10% to 90% for evaluating performance with sparse and dense data, respectively. The horizontal axis of each figure is the number of recommendations (K) that varies from 5 to 50. The vertical axis depicts the obtained recall in percentage.

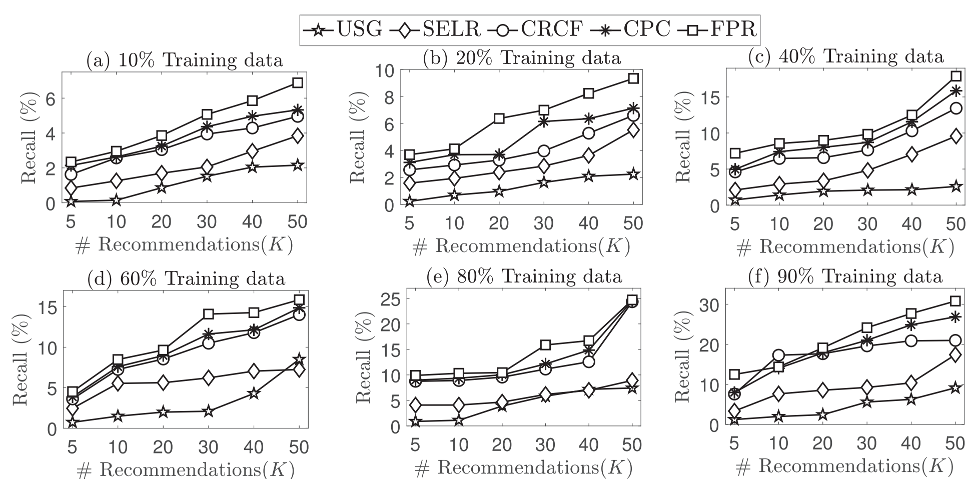


Fig. 14. Recall of the various POI recommendation approaches for the Pennsylvania (PA) state. The training set is varied from 10% to 90% for evaluating performance with sparse and dense data, respectively. The horizontal axis of each figure is the number of recommendations (K) that varies from 5 to 50. The vertical axis depicts the obtained recall in percentage.

Arizona (dense) compared to 11.84% for Karlsruhe (sparse) with 90% training and 50 number of recommendations. This is due to the fact that availability of wide range of locations within a state helps to understand a user's preference and also in correlating them with potential candidate POIs. However, the proposed FRP provides 11.84% accuracy compared to 10.3%, 10%, 5.3%, and 3% accuracy of CPC, CRCF, SELR, and USG recommender systems, respectively, at $K = 50$ and 90% training data for Karlsruhe. It has also been observed that

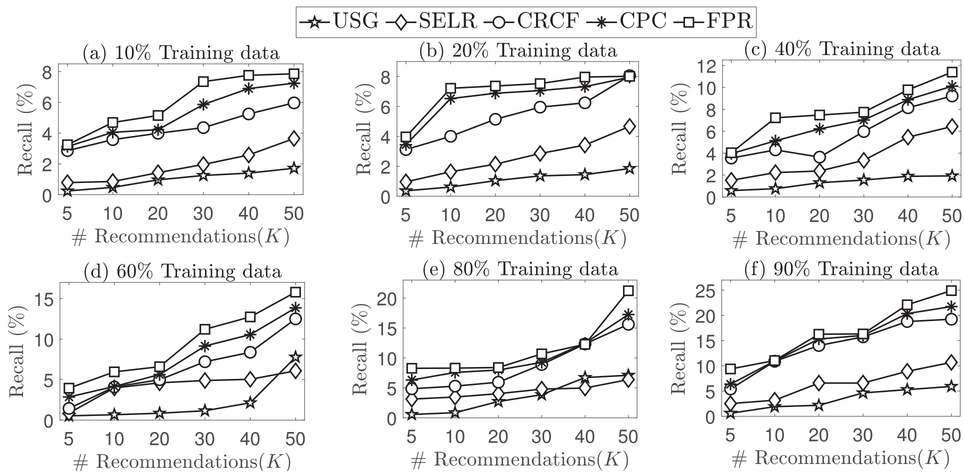


Fig. 15. Recall of the various POI recommendation approaches for the Edinburgh (EDH) state. The training set is varied from 10% to 90% for evaluating performance with sparse and dense data, respectively. The horizontal axis of each figure is the number of recommendations (K) that varies from 5 to 50. The vertical axis depicts the obtained recall in percentage.

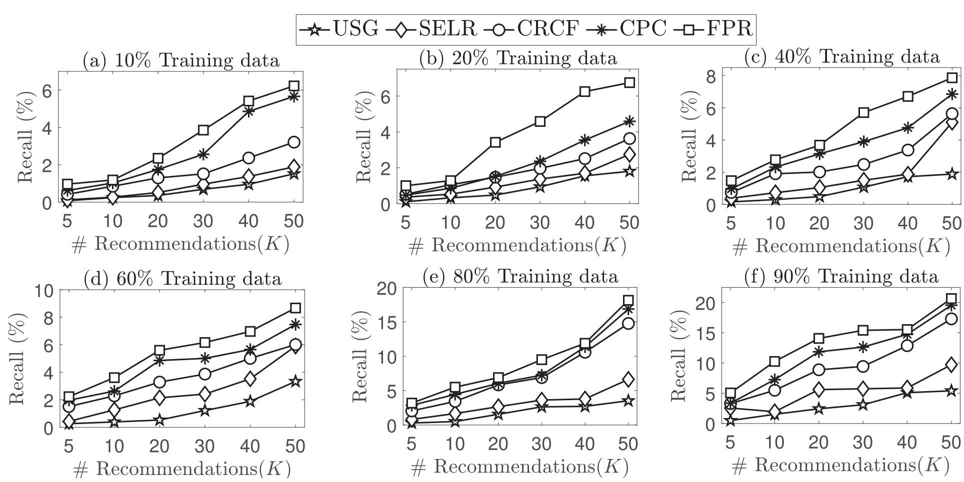


Fig. 16. Recall of the various POI recommendation approaches for the Wisconsin (WI) state. The training set is varied from 10% to 90% for evaluating performance with sparse and dense data, respectively. The horizontal axis of each figure is the number of recommendations (K) that varies from 5 to 50. The vertical axis depicts the obtained recall in percentage.

for all the competitive POI recommendation approaches, the accuracy improves with more number of recommendations. Further, we also provide the overall accuracy of the proposed approach when compared with other existing recommendation approaches for the overall working dataset (8 states combined). Figure 18 shows the obtained accuracy (in %) of the compared recommendation approaches. A similar tendency is observed on increasing the density of training set; performance of all POI recommenders is improved. Results for all the individual states combined show that the FPR approach significantly outperforms other POI recommendation approaches.

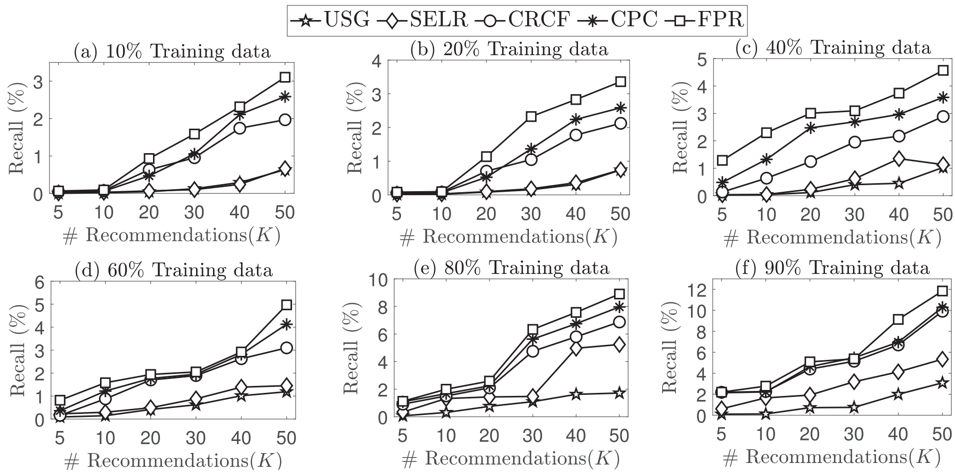


Fig. 17. Recall of the various POI recommendation approaches for the Karlsruhe (BW) state. The training set is varied from 10% to 90% for evaluating performance with sparse and dense data, respectively. The horizontal axis of each figure is the number of recommendations (K) that varies from 5 to 50. The vertical axis depicts the obtained recall in percentage.

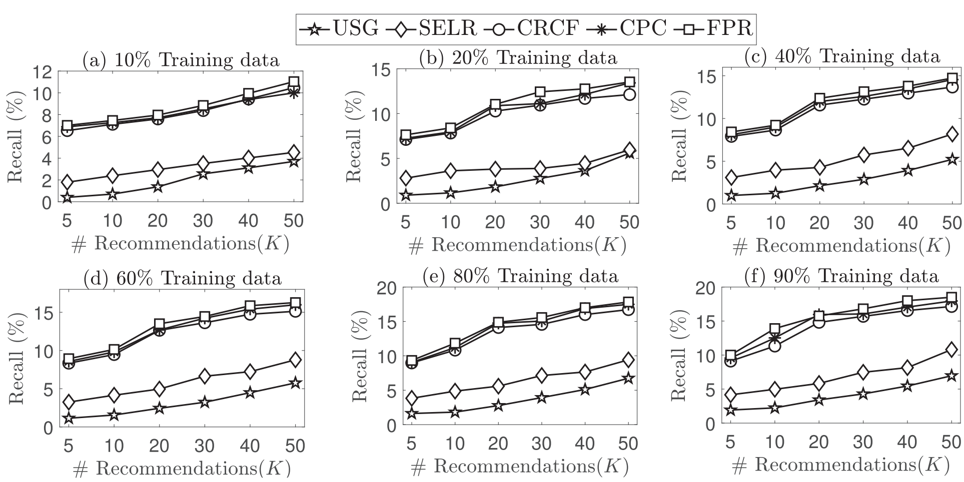


Fig. 18. Recall of the various POI recommendation approaches for all the eight states combined as a single geographical location. The training set is varied from 10% to 90% for evaluating performance with sparse and dense data, respectively. The horizontal axis of each figure is the number of recommendations (K) that varies from 5 to 50. The vertical axis depicts the obtained recall in percentage.

ii. Ranking of a Relevant POI in the Recommended Lists:

Ranking of relevant POIs (ground truths) in the recommended lists can be evaluated using the mean reciprocal rank (MRR) metric [4]. It estimates how a relevant POI is placed in the recommended list. In best case, the relevant POI is found at the 1st top-K recommended POI list. MRR can be computed as,

$$MRR = \frac{1}{|U|} \sum_{i \in U} \frac{1}{|G_i|} \sum_{j \in G_i} \begin{bmatrix} \frac{1}{rank_j}; \, if \ \{j \in G_i\} \cap RL \neq \emptyset \\ 0 \ ; \, otherwise \end{bmatrix}, \tag{10}$$

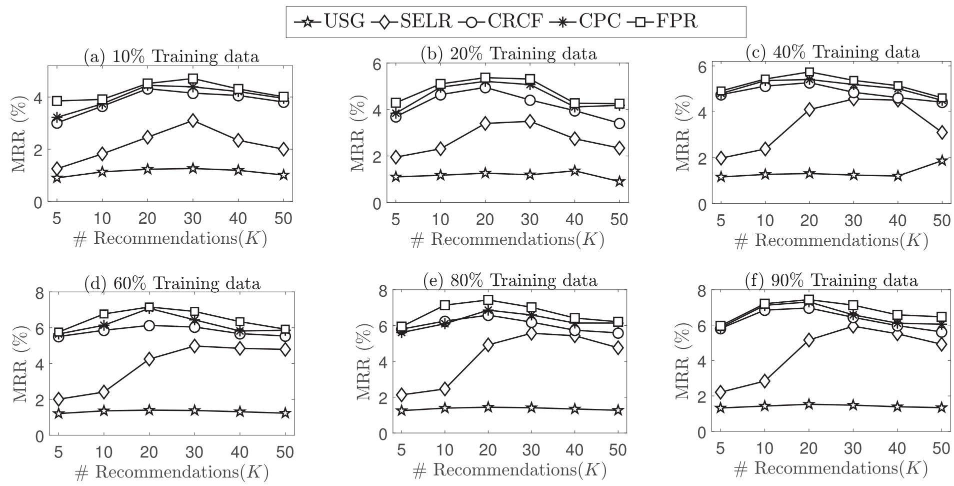


Fig. 19. MRR of various POI recommendation approaches for all the eight states combined as a single geographical location. The training set is varied from 10% to 90% for evaluating performance of the existing recommender systems with sparse and dense data, respectively. The horizontal axis of each figure is the number of recommendations (K) that varies from 5 to 50. The vertical axis depicts the obtained MRR in percentage.

where G_i is the set of all relevant POIs in the test dataset for user i. The rank of the POI j in the recommended list RL is depicted as $rank_j$.

The USG approach ranks the POIs that are having high probability of getting visited by the target user. SELR selects the top-K POIs with high rating by the target user. Both the approaches do not consider the geographical influence in recommendation. Therefore, the MRR for USG and SELR are lower than CRCF and the proposed FPR approach (Figure 19). However, with less historical data (20%–40%) it is observed that our approach supersedes CPC and CRCF in ranking the relevant POIs.

5.4.2 *Handling CSPs.*

i. Collecting CSPs for Yelp network:

   We execute the proposed CPISDR (Algorithm 2) for each of the eight selected states. The existing non-cold-start POIs for any state (say, Arizona, AZ) is taken as input $\mathcal{D}_{\mathcal{AZ}}$ in CPISDR and the probable cold-start POIs CSP_{AZ} for the Yelp network are collected by crowdsourcing the Foursquare and Google places network. The state-wise statistics of the CSPs for Yelp network are presented in Table 2. It is observed from the number of CSPs identified for Yelp that fusing different social networks can help in collecting new or cold-start POIs. Subsequently, these CSPs can also be recommended to target users.

ii. Successful identification of the marked CSPs in Yelp:

   The POIs in Yelp after removing the marked CSPs are provided as input $\mathcal{D}_{\mathcal{AZ}}$ to the proposed CPISDR. This strategy is followed for all the eight states. Next, the crowdsourced CSPs are verified with the marked CSPs in the test dataset of Yelp. The number of CSPs identified for Yelp network are used to compute the success rate. Existing approaches [49, 52, 55] do not specifically look into the task of identifying a cold-start POI. Therefore, in this work, we are unable to compare our proposed FPR approach with the existing techniques. Table 3 shows the performance of FPR over all the eight POI datasets for finding

Table 2. Results of Identifying Cold-start POIs for Yelp by Crowdsourcing Different Social Networks Such as Foursquare and Google Places

Area	#POIs in Yelp	#CSPs for Yelp by crowdsourcing
Arizona, US	32,615	225
Nevada, US	21,233	278
North Carolina, US	6,162	203
Quebec, CAN	4,943	154
Pennsylvania, US	3,754	187
Edinburgh, UK	3,206	139
Wisconsin, US	2,802	216
Karlsruhe, Germany	1,048	140

The results are provided for specific geographical area of eight states. #POIs in Yelp denotes the number of POIs in Yelp recorded in a state. #CSPs for Yelp are the new POIs in the corresponding area that are identified for Yelp network by crowdsourcing Foursquare and Google places.

Table 3. Experimental Results of Identifying the Marked CSPs of Yelp Dataset, for Each of the Eight States

Area	#POIs	#CSPs	#CSPs recovered	Success %
Arizona, US	32,615	3,261	2,869	88
Nevada, US	21,233	2,123	1,804	85
North Carolina, US	6,162	616	455	74
Quebec, CAN	4,943	494	360	73
Pennsylvania, US	3,754	375	307	82
Edinburgh, UK	3,206	320	204	64
Wisconsin, US	2,802	280	178	64
Karlsruhe, Germany	1,048	104	57	55

#POIs denotes the number of recorded POIs observed in the Yelp dataset of the corresponding state. #CSPs are the marked cold-start POIs of Yelp dataset. #CSPs recovered are the number of cold-start POIs that are re-identified for Yelp by crowdsourcing other social networks. Success % is the percentage of marked CSPs recovered for Yelp.

their corresponding cold-start POIs. Arizona being the most popular region in Yelp attracts more entries through online social media. This is clearly visible that the percentage of successful CSP identifications in Arizona is more than Quebec, where most of the entries are found from the city of Montreal. Crowdsourcing mostly depends upon the contents of the target social networks (Foursquare and Google places, in this case). Therefore, combining more number of social networks can increase the chances of identifying large number of CSPs. We expect a higher success rate in identifying the CSPs for Yelp, if more number of social networks are combined.

iii. Accuracy in recommending the marked CSPs in Yelp:

Here, we evaluate whether the proposed FPR approach can actually recommend the cold-start POIs. A similar strategy is adopted as mentioned in Section 5.4.1(ii). Only difference is that here the training dataset is generated by removing the CSPs from each user's historical data. The recall metric mentioned in Equation (9) is used for evaluation. In this case, $Ground\ truths_i$ are the CSPs in the test dataset for user i. Figure 20 depicts the recall (in %) of the existing recommender systems on all the individual eight states, respectively. Similar to the previous experiments, the number of recommendations (K) is varied from 5 to 50. Next, we introduce a strategy to evaluate importance of the cross-domain knowledge

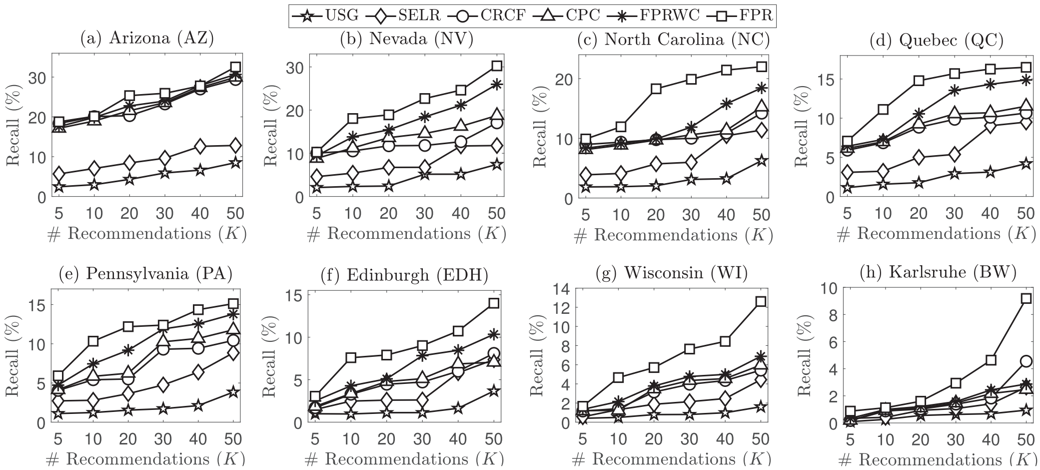


Fig. 20. Recall of various POI recommender systems in handling the CSPs for each of the eight states are depicted here. The horizontal axis of each figure is the number of recommendations (K) that varies from 5 to 50. The vertical axis depicts the obtained recall in percentage.

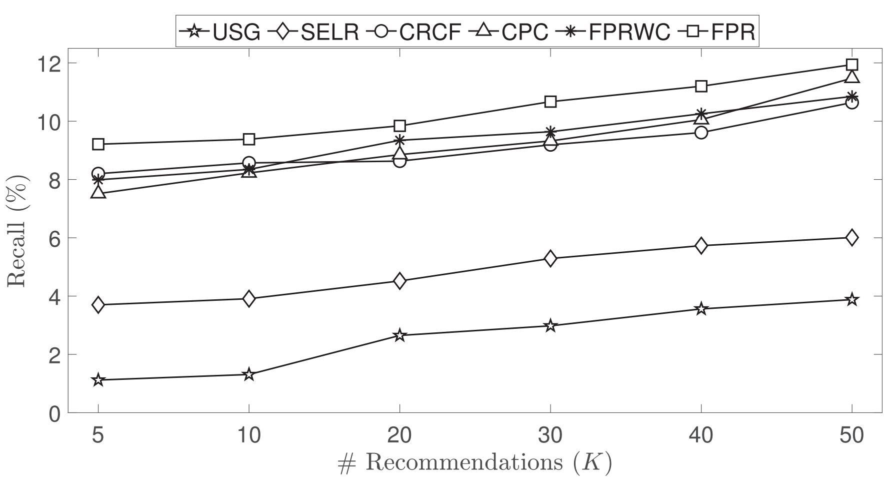


Fig. 21. Recall of the compared POI recommender systems in handling the CSPs for all the eight states combined as a single geographical location. The horizontal axis of each figure is the number of recommendations (K) that varies from 5 to 50. The vertical axis depicts the obtained recall in percentage.

in our proposed recommender system. The proposed approach is evaluated both with and without the cross-domain knowledge. Our approach without the cross-domain knowledge is depicted as FPRWC in the rest of this article. Highest accuracy (32.56%) is observed in Arizona with $K = 50$ recommendations by the proposed FPR. The results obtained for the existing CRCF approach are found to be closest to our results among the compared POI recommendation approaches. Even our approach without cross-domain knowledge has lower performance with respect to the baseline CRCF with less number of recommendations (K). This tendency is observed for almost all the datasets with dense data. However, for all the geographical locations the proposed FPR with the cross-domain knowledge outperforms the existing recommender systems. Figure 21 shows the recall of our approach

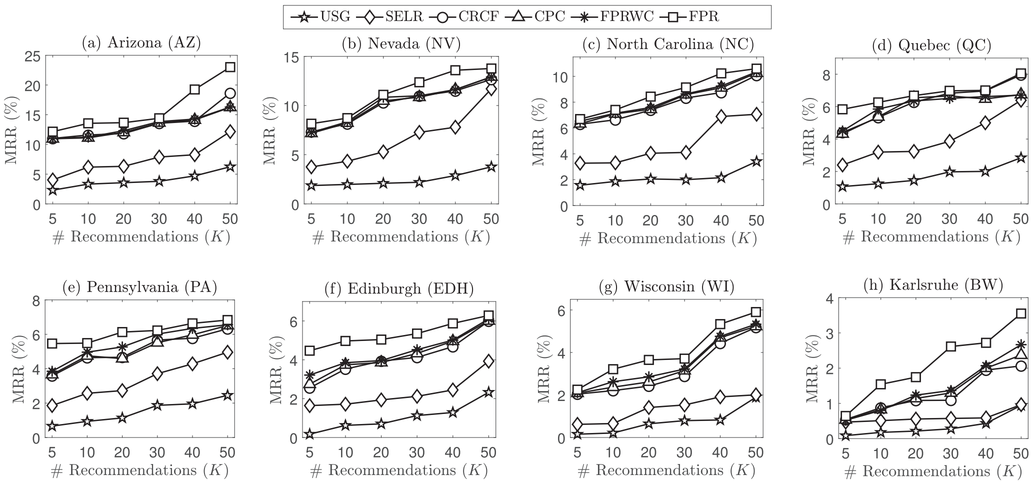


Fig. 22. Mean reciprocal rank (MRR) of various POI recommender systems in handling the CSPs for each of the eight states are depicted here. The horizontal axis of each figure is the number of recommendations (K) that varies from 5 to 50. The vertical axis depicts the obtained recall in percentage.

and the existing state-of-the-art techniques for all the eight states combined as a single geographical location. Crowdsourcing review data at a CSP from various online sources helps to gather a wide range of aspects. This is observed from the better accuracy of FPR when compared to others in recommending the CSPs. However, it is also observed that the accuracy of recommendations increases for all the approaches with more number of recommendations.

iv. Ranking of the marked CSPs in the recommended lists:

As mentioned earlier, the best performance of a ranked recommended list is observed when the relevant CSP is placed at position 1 of the recommended list. To evaluate performance of the proposed recommender system in ranking the marked CSPs of Yelp in the recommended list, we utilize the MRR metric as mentioned in Equation (10). Here, G_i is the set of all marked CSPs in the test dataset for user i and $rank_j$ is the rank of the CSP j in the recommended list RL. Figure 22 depicts the MRR (%) of the recommender systems for all the individual eight states. In addition to this, we also provide MRR of the eight states when combined as a single geographical area in Figure 23. Here, we observe the variation in ranking the CSPs as more number of recommendations are performed. Increase in the number of recommendations does not necessarily improve the rank of a relevant CSP. The best MRR is achieved at K as 10, 40, 20, 50, 50, and 10 for USG, SELR, CRCF, CPC, FPRWC, and proposed FPR, respectively (Figure 23). Number of recommendations within 10 to 20 POIs are found to provide best rankings to the CSPs for our proposed FPR approach.

6 CONCLUSIONS AND FUTURE WORK

Handling newly evolved POIs (cold-start POIs) is an important issue in POI recommender system. In this article, we presented a POI recommendation approach that can identify cold-start POIs in a region and combine them along with other non-cold-start POIs for relevant recommendations to an active user. The cold-start POIs are found by crowdsourcing review and rating data from various online social networks. The cross-domain knowledge at the cold-start POIs are then exploited to learn the inherent features and the implicit ratings. Subsequently, a feature-based similarity

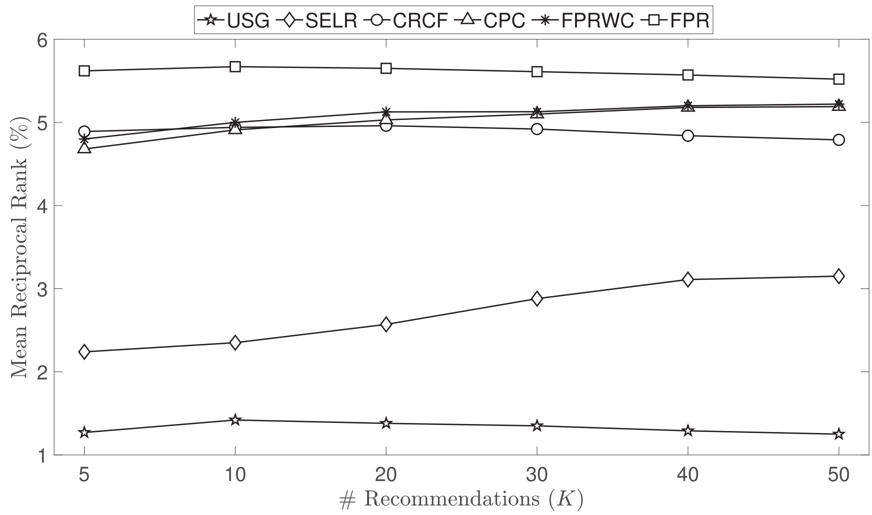


Fig. 23. Mean reciprocal rank (MRR) of the compared POI recommender systems in handling the CSPs for all the eight states combined as a single geographical location. The horizontal axis of each figure is the number of recommendations (K) that varies from 5 to 50. The vertical axis depicts the obtained recall in percentage.

measure is used to compute the nearest cluster of POIs for an active user. Finally, a collaborative filtering approach is combined with the geographical influence for recommending the top-K POIs.

This work can be extended in four directions as follow: First, time factor can be incorporated in POI recommendations. The set of top-K relevant POIs needs to be categorized on the basis of the time interval of a day. For a POI, certain particular features such as the opening and closing hours can be explored in this regard. Second, the current system needs to be updated after a predefined period of time for finding the cold-start POIs. Developing a system that is capable of continuous monitoring of specific social networks can help in reducing periodic updates. Moreover, such a system should be more robust in handling cold-start POIs. Third, we can perform a subjective analysis on how the cold-start POIs of a city are perceived by the cold-start users of the city. Finally, if a cold-start POI is not available in any social network, then such cold-start POI is never considered by FPR for recommendation. Therefore, even though a cold-start POI is very relevant with respect to a target user, the POI can never be recommended if it is not present in any social network. In this direction, we can utilize the neighboring non-cold-start points-of-interest for estimating the probability of the target user to visit the cold-start point-of-interest.

REFERENCES

[1] Wouter Bancken, Daniele Alfarone, and Jesse Davis. 2014. Automatically detecting and rating product aspects from textual customer reviews. In *Proceedings of the 1st International Workshop on Interactions between Data Mining and Natural Language Processing at ECML/PKDD*. 1–16.

[2] Betim Berjani and Thorsten Strufe. 2011. A recommendation system for spots in location-based online social networks. In *Proceedings of the 4th Workshop on Social Network Systems*. ACM, 4:1–4:6.

[3] James C. Bezdek, Robert Ehrlich, and William Full. 1984. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* 10, 2–3 (1984), 191–203.

[4] Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. 2006. Learning to rank with nonsmooth cost functions. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*. The MIT Press, 193–200.

[5] Fidel Cacheda, Víctor Carneiro, Diego Fernández, and Vreixo Formoso. 2011. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Trans. Web* 5, 1 (2011), 2:1–2:33.

[6] Wen-Yen Chen, Jon-Chyuan Chu, Junyi Luan, Hongjie Bai, Yi Wang, and Edward Y. Chang. 2009. Collaborative filtering for Orkut communities: Discovery of user latent behavior. In *Proceedings of the 18th International Conference on World Wide Web*. ACM, 681–690.

[7] Qiang Cui, Shu Wu, Qiang Liu, Wen Zhong, and Liang Wang. 2018. MV-RNN: A multi-view recurrent neural network for sequential recommendation. *IEEE Trans. Knowl. Data Eng.* 32, 2 (2018), 317–331.

[8] Mukund Deshpande and George Karypis. 2004. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.* 22, 1 (2004), 143–177.

[9] Andrea Esuli and Fabrizio Sebastiani. 2007. SentiWordNet: A high-coverage lexical resource for opinion mining. *Evaluation* 17 (2007), 1–26.

[10] Gregory Ference, Mao Ye, and Wang-Chien Lee. 2013. Location recommendation for out-of-town users in location-based social networks. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. ACM, 721–726.

[11] Huiji Gao, Jiliang Tang, and Huan Liu. 2012. gSCorr: Modeling geo-social correlations for new check-ins on location-based social networks. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. ACM, 1582–1586.

[12] Huiji Gao, Jiliang Tang, and Huan Liu. 2015. Addressing the cold-start problem in location recommendation using geo-social correlations. *Data Mining Knowl. Disc.* 29, 2 (2015), 299–323.

[13] Thomas George and Srujana Merugu. 2005. A scalable collaborative filtering framework based on co-clustering. In *Proceedings of the 5th International Conference on Data Mining*. IEEE, 625–628.

[14] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 230–237.

[15] Bo Hu and Martin Ester. 2014. Social topic modeling for point-of-interest recommendation in location-based social networks. In *Proceedings of the International Conference on Data Mining*. IEEE, 845–850.

[16] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 168–177.

[17] Shangrong Huang, Jian Zhang, Lei Wang, and Xian-Sheng Hua. 2016. Social friend recommendation based on multiple network correlation. *IEEE Trans. Multimedia* 18, 2 (2016), 287–299.

[18] Yogesh Jhamb and Yi Fang. 2017. A dual-perspective latent factor model for group-aware social event recommendation. *Inf. Proc. Manag.* 53, 3 (2017), 559–576.

[19] Zhipeng Jin, Qiudan Li, Daniel D. Zeng, YongCheng Zhan, Ruoran Liu, Lei Wang, and Hongyuan Ma. 2016. Jointly modeling review content and aspect ratings for review rating prediction. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 893–896.

[20] Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*. Association for Computational Linguistics, 1–7.

[21] Ryan Kiros, Yukun Zhu, Ruslan R. Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*. 3294–3302.

[22] Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *J. ACM* 46, 5 (1999), 604–632.

[23] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction. In *Proceedings of the International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2341–2347.

[24] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[25] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 617–624.

[26] Huayu Li, Yong Ge, Richang Hong, and Hengshu Zhu. 2016. Point-of-interest recommendations: Learning potential check-ins from friends. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 975–984.

[27] Defu Lian, Yong Ge, Fuzheng Zhang, Nicholas Jing Yuan, Xing Xie, Tao Zhou, and Yong Rui. 2015. Content-aware collaborative filtering for location recommendation based on human mobility data. In *Proceedings of the IEEE International Conference on Data Mining*. IEEE, 261–270.

[28] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synth. Lect. Hum. Lang. Technol.* 5, 1 (2012), 1–167.

[29] Nathan N. Liu, Xiangrui Meng, Chao Liu, and Qiang Yang. 2011. Wisdom of the better few: Cold start recommendation via representative based rating elicitation. In *Proceedings of the 5th ACM Conference on Recommender Systems*. ACM, 37–44.

[30] Shenghao Liu, Bang Wang, and Minghua Xu. 2017. Event recommendation based on graph random walking and history preference reranking. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 861–864.

[31] Hao Ma, Irwin King, and Michael R. Lyu. 2007. Effective missing data prediction for collaborative filtering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 39–46.

[32] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations.* 55–60.

[33] Pramit Mazumdar, Bidyut Kr. Patra, and Korra Sathya Babu. 2018. Handling cold-start scenarios in point-of-interest recommendations through crowdsourcing. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data.* ACM, 322–324.

[34] Andriy Mnih and Ruslan R. Salakhutdinov. 2008. Probabilistic matrix factorization. In *Proceedings of the Conference on Advances in Neural Information Processing Systems.* 1257–1264.

[35] Tieyun Qian, Bei Liu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2019. Spatiotemporal representation learning for translation-based POI recommendation. *ACM Trans. Inf. Syst.* 37, 2 (2019), 18.

[36] Jitendra Kumar Rout, Smriti Singh, Sanjay Kumar Jena, and Sambit Bakshi. 2017. Deceptive review detection using labeled and unlabeled data. *Multimedia Tools Applic.* 76, 3 (2017), 3187–3211.

[37] Gerard Salton and Michael J. McGill. 1986. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc.

[38] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2000. *Application of Dimensionality Reduction in Recommender System—A Case Study.* Technical Report. DTIC Document. WebKDD-2000 Workshop.

[39] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web.* ACM, 285–295.

[40] Bernard W. Silverman. 2018. *Density Estimation for Statistics and Data Analysis.* Routledge.

[41] Gamgarn Somprasertsri and Pattarachai Lalitrojwong. 2010. Extracting product features and opinions from product reviews using dependency analysis. In *Proceedings of the 7th International Conference on Fuzzy Systems and Knowledge Discovery*, Vol. 5. IEEE, 2358–2362.

[42] Kaisong Song, Wei Gao, Shi Feng, Daling Wang, Kam-Fai Wong, and Chengqi Zhang. 2017. Recommendation vs sentiment analysis: A text-driven latent factor model for rating prediction with cold-start awareness. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence.* AAAI Press, 2744–2750.

[43] Kaisong Song, Wei Gao, Lujun Zhao, Jun Lin, Changlong Sun, and Xiaozhong Liu. 2019. Cold-start aware deep memory network for multi-entity aspect-based sentiment analysis. In *Proceedings of the International Joint Conference on Artificial Intelligence.* AAAI Press, 5197–5203.

[44] Lijun Tang and Eric Yi Liu. 2017. Joint user-entity representation learning for event recommendation in social network. In *Proceedings of the IEEE 33rd International Conference on Data Engineering.* IEEE, 271–280.

[45] Yi Wang and Xiao-Jing Wang. 2005. A new approach to feature selection in text classification. In *Proceedings of the International Conference on Machine Learning and Cybernetics.* IEEE, 3814–3819.

[46] Zhibo Wang, Jilong Liao, Qing Cao, Hairong Qi, and Zhi Wang. 2015. Friendbook: A semantic-based friend recommendation system for social networks. *IEEE Trans. Mob. Comput.* 14, 3 (2015), 538–551.

[47] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. 2016. Learning graph-based POI embedding for location-based recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.* ACM, 15–24.

[48] Jingwei Xu, Yuan Yao, Hanghang Tong, Xianping Tao, and Jian Lu. 2015. Ice-breaking: Mitigating cold-start recommendation problem by rating comparison. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence.* ACM, 3981–3987.

[49] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. 2013. A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media.* ACM, 119–128.

[50] Jun Yang, Runqi Yang, Chongjun Wang, and Junyuan Xie. 2018. Multi-entity aspect-based sentiment analysis with context, entity and aspect memory. In *Proceedings of the International Joint Conference on Artificial Intelligence.* AAAI Press, 6029–6036.

[51] Mao Ye, Peifeng Yin, and Wang-Chien Lee. 2010. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems.* ACM, 458–461.

[52] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 325–334.

[53] Hongzhi Yin, Xiaofang Zhou, Bin Cui, Hao Wang, Kai Zheng, and Quoc Viet Hung Nguyen. 2016. Adapting to user interest drift for POI recommendation. *IEEE Trans. Knowl. Data Eng.* 28, 10 (2016), 2566–2581.

[54] Fei Yu, Nan Che, Zhijun Li, Kai Li, and Shouxu Jiang. 2017. Friend recommendation considering preference coverage in location-based social networks. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 91–105.

[55] Chenyi Zhang and Ke Wang. 2016. POI recommendation through cross-region collaborative filtering. *Knowl. Inf. Syst.* 46, 2 (2016), 369–387.

[56] Jia-Dong Zhang and Chi-Yin Chow. 2013. iGSLR: Personalized geo-social location recommendation: A kernel density estimation approach. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 334–343.

[57] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 83–92.

[58] Yu Zheng and Xing Xie. 2011. Learning travel recommendations from user-generated GPS traces. *ACM Trans. Intell. Syst. Technol.* 2, 1 (2011), 2:1–2:29.

Received May 2018; revised December 2019; accepted June 2020