information

MDPI

Article

Recommendation System Algorithms on Location-Based Social Networks: Comparative Study

Abeer Al-Nafjan [1,*], Norah Alrashoudi [2] and Hend Alrasheed [2]

[1] Department of Computer Science, College of Computer and Information Sciences, Imam Muhammad Bin Saud University, Riyadh 11432, Saudi Arabia
[2] Department of Information Technology, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia; 442203043@student.ksu.edu.sa (N.A.); halrasheed@ksu.edu.sa (H.A.)
* Correspondence: annafjan@imamu.edu.sa; Tel.: +966-112-597-570

Abstract: Currently, social networks allow individuals from all over the world to share ideas, activities, events, and interests over the Internet. Using location-based social networks (LBSNs), users can share their locations and location-related content, including images and reviews. Location rec-14 recommendation system-based LBSN has gained considerable attention in research using techniques and methods based on users' geosocial activities. In this study, we present a comparative analysis of three matrix factorization (MF) algorithms, namely, singular value decomposition (SVD), singular value decomposition plus (SVD++), and nonnegative matrix factorization (NMF). The primary task of the implemented recommender system was to predict restaurant ratings for each user and make a recommendation based on this prediction. This experiment used two performance metrics for evaluation, namely, root mean square error (RMSE) and mean absolute error (MAE). The RMSEs confirmed the efficacy of SVD with a lower error rate, whereas SVD++ had a lower error rate in terms of MAE.

Keywords: recommendation system; location-based social networks (LBSNs); matrix factorization (MF); points of interest (POIs)

Citation: Al-Nafjan, A.; Alrashoudi, N.; Alrasheed, H. Recommendation System Algorithms on Location-Based Social Networks: Comparative Study. *Information* **2022**, *13*, 188. https://doi.org/10.3390/info13040188

Academic Editors: Marco Polignano and Giovanni Semeraro

Received: 20 February 2022
Accepted: 6 April 2022
Published: 8 April 2022

1. Introduction

Currently, recommendation systems play an important role in most commercial applications. These systems help increase both users' satisfaction and content providers' revenue by capturing users' personalized preferences. The central concept of recommendation systems is personalized prediction, to which the most common approach is collaborative filtering (CF), which employs users' interactions and interests to define their preferences among a considerable number of items [1].

In location-based social networks (LBSNs) such as Foursquare, Yelp, and TripAdvisor, users tend to share their experiences of places with others; moreover, they use others' ratings and reviews to determine which places to visit. These networks are rapidly growing, with huge volumes of different types of data. This increases the complexity of users selecting points of interests (POIs), such as restaurants, hotels, shops, tourist attractions, and cultural sites.

Many factors affect users' POIs, including geolocation, user preferences, and social influences [2]. Therefore, the primary function of POI-recommendation systems offered by LBSN services is to recommend new POIs relevant to the user, thereby helping them explore new places and know their cities better [2,3]. Recommendation results are often biased toward popular places with higher ratings and ignore those that are less popular yet may be relevant to user preferences [2]. Moreover, the influence of social networks encourages users to explore new POIs and improves recommendation accuracy in systems using CF. Certain studies have proposed recommendation systems that use CF to take into account social and geographical influences [3–5].

Two common challenges affect the implementation of a recommendation system using CF. The first consideration is the size of the processed datasets. The second stems from the sparseness of the rating matrix, which means that only a small number of items are rated for each user. Matrix factorization–based CF has effectively addressed these issues.

The matrix factorization (MF) method is an unsupervised learning method for decomposing latent variables and reducing dimensionality. It is one of the most widely used techniques in the fields of intelligent systems and data mining. It has proven to be effective in a variety of real-world applications, including recommender systems. Several studies [6–9] have stated that MF is used specifically for processing large recommendation systems' databases and providing scalable approaches.

Due to the considerable amount of data in social networks and the complexity of predicting relevant recommendations for users, different MF methods have been developed to enhance the accuracy of recommender systems. In this paper, we propose a recommendation system based on the idea that information extracted from available ratings affects unknown ratings. The implemented recommender system's primary task is to predict each user's restaurant ratings and make recommendations based on this prediction.

To this end, we investigated the performance of three MF methods, namely singular value decomposition (SVD), singular value decomposition plus (SVD++), and nonnegative matrix factorization (NMF), which are often used in the CF of recommendation systems. Two performance metrics, namely, root mean square error (RMSE) and mean absolute error (MAE), will be used to evaluate the performance of these methods. This study tends to evaluate and analyze the three algorithms to help researchers and practitioners identify efficient solutions in recommendation systems.

The remainder of this study is organized as follows. Section 2 provides the primary concepts of the recommendation systems and their techniques. Section 3 presents a description of the investigated algorithms. Section 4 discusses the experiment's dataset, evaluation metrics, and setup. Section 5 compares the performance of the selected algorithms. Finally, Section 6 provides the conclusion of this study.

2. Background

2.1. Location-Based Social Networks

Recently, the popularity of online social networks (e.g., Facebook) has increased to over 1.5 billion users [10]. These social networks allow people to communicate, interact, and share content with other users via online platforms. Different types of online social networks exist, based on the type of content shared on each platform. For example, Facebook (https://www.facebook.com, accessed date 2 February 2022). and Twitter (https://www.Twitter.com, accessed date 2 February 2022) are based on short text content; thus, they are considered micro-blogging networks. By contrast, other social networks such as YouTube (https://www.youtube.com, accessed date 2 February 2022) and Flickr (https://www.flickr.com, accessed date 2 February 2022) depend on multimedia content.

Given the availability of location-acquisition technologies (e.g., GPS), social networks function with location data in different ways. For example, users can share their current location on a website, upload location-tagged photos on a social network platform, or write a review for a place that they visited. Therefore, location is considered one of the primary important components of user context, and extensive knowledge about users' behavior and interests can be derived from their location data [11].

These social networks are LBSNs, each comprising two social structures: one is a social network that adds location information to let people share location-embedded content; the other is a social network that connects people by their immediate locations in the physical world via location-tagged media such as images, videos, and text. The user's physical location is represented either as a current location at a given timestamp or a history of a location that a user visited at a certain time.

For an LBSN's recommendation system, POI recommendation is an important task aimed at providing users with personalized places, to help them explore their preferences.

In recent years, POI recommendation has utilized different techniques and methods to produce content-based recommendations, CF-based recommendations, or a hybrid of the two. In the second and third sections, we discuss different filtering techniques and related works.

2.2. Content-Based POI Recommendation

In general, the content-based filtering approach recommends items based on their similarity to items that the user previously liked, so user profiles and item descriptions are important elements in the process.

In content-based POI recommendation, user behavioral data (e.g., ratings) are used as important features for the recommendation process. Recently, many studies have used content-based recommendation by analyzing users' ratings and predicting ratings for unvisited POIs [12,13]. Data scarcity is one of the primary issues in POI recommendation, and many researchers are addressing this issue by adding more contextual information (e.g., temporal data) to enhance recommendation performance [14–17].

Ma et al. [12] proposed a POI recommendation based on the LBSN framework. Four data types were combined: user check-in information, user rating information, geographic influence of POIs, and similarity of POIs. The experiment was conducted using the Gowalla and Foursquare datasets. The authors proposed an extended user preference model based on matrix factorization (UPEMF) as a recommendation-algorithm model. The results show that the UPEMF achieves higher accuracy than other recent methods such as probabilistic MF and geographical and user MF do.

Wang et al. [13] aimed to use a deep learning model based on a recurrent neural network (RNN). They proposed a similarity-based POI-embedding and recurrent neural network with temporal influence (SPENT). Based on the embedding vectors, they organized the POIs into a similarity tree. The experiment was conducted using the Gowalla and Foursquare datasets. The experimental results show that SPENT outperformed the other prior methods in terms of precision and recall.

Zhao et al. [14] proposed a spatial-temporal latent ranking for successive point-of-interest recommendation (STELLAR) system that aimed to capture the impact of time on successive POI recommendations. Their research findings show that the STELLAR model outperformed state-of-the-art models in their experiments on two datasets, Foursquare and Gowalla.

Zhao et al. [15] proposed a spatiotemporal gated network (STGN) for POI recommendation, based on a gated mechanism that enhanced the long short-term memory (LSTM)–based RNN network by introducing gates to capture the spatiotemporal relationships between successive check-ins.

Qian et al. [16] used knowledge graph embedding by modeling the third-order relationship among users, POIs, and spatiotemporal contexts for large-scale POI recommendations. Furthermore, Cai et al. [17] have addressed data scarcity and implicit feedback by integrating spatial and temporal features in a factorization model and using a hybrid optimization algorithm that combines the weighted MF and ranking model. Table 1 summarizes previous studies on content-based POI recommendation.

Table 1. Summary of previous studies on content-based POI recommendation.

Content-Based Recommendation	Ref.	Title
Rating prediction	[12]	Multifactor Fusion POI Recommendation Model
	[13]	PENT: A Successive POI Recommendation Method Using Similarity-Based POI Embedding and Recurrent Neural Network with Temporal Influence.
Temporal-based	[14]	STELLAR: Spatial-Temporal Latent Ranking for Successive Point-of-Interest Recommendation.
	[15]	Where to Go Next: A Spatio-Temporal Gated Network for Next POI Recommendation.
	[16]	Spatiotemporal Representation Learning for Translation-Based POI Recommendation.
	[17]	Integrating Spatial and Temporal Contexts into a Factorization Model for POI Recommendation.

2.3. Collaborative Filtering-Based POI Recommendation

In CF, users' check-in data are used to extract their preferences concerning POIs [4]. The traditional CF algorithms comprise two primary categories, namely memory- and model-based CF recommendation algorithms [18]. The memory-based CF algorithm can be divided further into two subcategories: user- and item-based CF recommendation algorithms. In the user-based CF, users' preferences are discovered by aggregating the behavior data of other similar users [3].

Many studies have proposed user-based CF for POI recommendation systems. Jia et al. [19] developed a user-based CF for a tourist attraction recommender system. The recommender system was developed as an online application capable of producing a personalized preferred attraction for tourists by using the cosine method to calculate similarities between users. Certain researchers combined social and geographical influences by developing user-based CF to explore POIs through user check-in activity in LBSNs [3]. Moreover, Zhang et al. [4] incorporated user-based CF, memory-based attention to users' preferences, and POI stickiness to improve the performance of POI recommendations.

Machine-learning algorithms have also been used for user-based CF and POI recommendation. For example, Yang et al. [1] used neural networks that combined CF and semi-supervised learning (SSL) to model nonlinear interactions among users and POIs. In terms of item-based CF, the user-item relevance was discovered using the degree of similarity between a target item and items the user interacted with, based on historical data [20]. Ye et al. [3] reported that item-based CF (in comparison to user-based CF) is not an effective approach for discovering POIs in LBSNs because many POIs have few user check-ins.

Zhang et al. [4] proposed a user-CF-memory-stickiness algorithm, a user-based CF framework with memory-based preference attenuation and POI stickiness for top-N POI recommendation tasks. Jia et al. [19] proposed a user-based tourist attraction recommender system based on the CF method.

However, model-based CF is applied in LBSNs to predict POIs of unseen items by using latent factors [21]. Jiang et al. [22] proposed the author topic model-based CF method that provides users planning to visit new cities with personalized POI recommendations by learning their travel preferences from text descriptions of their shared images on social media. Table 2 summarizes previous studies on CF POI recommendation systems.

Table 2. Summary of previous studies on collaborative filtering (CF)-based POI recommendation.

Collaborative Filtering	Ref.	Title
User-based	[3]	Exploiting geographical influence for collaborative point-of-interest recommendation.
	[4]	Incorporating Memory-Based Preferences and Point-of-Interest Stickiness into Recommendations in Location-Based Social Networks.
	[19]	User-Based Collaborative Filtering for Tourist Attraction Recommendations.
	[1]	Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation.
Model-based	[21]	Model-Based Collaborative Filtering for Recommender Systems: An Empirical Survey.
	[22]	Author Topic Model-Based Collaborative Filtering for Personalized POI Recommendation.

2.4. Matrix Factorization

CF-based recommender systems often represent ratings as a matrix of two entities: users and items. Due to the high-dimensional data in social networks and the complexity of extracting common features from these data, MF was proposed to deal with these problems. MF is defined as the decomposition of the initial matrix in two or more smaller metrics to reduce required storage space and processing time [6].

Multiple algorithms use the MF approach, and the most common ones are nonnegative matrix factorization (NMD), probabilistic matrix factorization, SVD, and SVD++. In this study, we will compare three of these algorithms: NMD, SVD, and SVD++. The first algorithm, NMD, decomposes the initial matrix into two smaller matrices, provided that each element of the decomposed matrices should be nonnegative. The second algorithm, SVD, is used in solving real and complex matrices, rate estimation, and canonical correlation analysis [23]. The third algorithm, SVD++, combines the strengths of the latent and neighbor models.

3. Description of the Investigated Algorithms

In this study, we aimed to examine and compare three algorithms, namely, NMD, SVD, and SVD++. The first algorithm is NMD, which decomposes the initial matrix into two smaller matrices, provided that each element of decomposed matrices is nonnegative. The second algorithm is SVD, which is used for solving real and complex metrics, rate estimation, and for canonical correlation analysis [23]. The third algorithm is SVD++, which combines the strengths of the latent and neighbor models.

3.1. Singular Value Decomposition (SVD)

The SVD technique is a MF technique that is commonly used in item-based recommender systems [24]. SVD confirms its efficiency in multiple recommender systems by resolving data sparsity issues that could reduce recommendation quality [24–26].

SVD involves extracting features and correlations from the user-item matrix. The SVD of an $(m \times n)$ matrix A produces features that factor a matrix A into the three matrices U, Σ, and V as shown in the following equation:

$$SVD(A) = U\Sigma V^T \tag{1}$$

where U and V are $(m \times m)$ and $(n \times n)$ orthogonal matrices, respectively, and Σ is the $(m \times n)$ singular orthogonal matrix with nonnegative elements.

The diagonal entries of Σ are called the singular values of A. The column vectors of U are called the left singular vectors, and the column vectors of V are called the right singular vectors.

In this study, we examined the use of SVD in a location-based recommendation system and evaluated its performance. The primary processes of SVD are described as follows. First, SVD linearly combines the relations between users and items via certain factors. Then, it induces a low-dimensional representation of the original user-item space and calculates the neighborhood in the reduced space. Finally, SVD generates a recommendation for the user as a list of top-N items [27].

3.2. Singular Value Decomposition Plus Plus

SVD++ is a type of MF method derived from SVD to provide personalized recommendations in CF. SVD++ adds a neighborhood method with a latent factor to its model, whereas SVD depends only on the latent factor method [28]. SVD++ has achieved improved predictive accuracy in many studies of different recommendation systems, including movie recommendations [28–30], tourist attractions [31], and consumer reviews of shopping websites [32].

SVD++ combines the neighborhood and latent factor approaches to improve prediction accuracy. The neighborhood model is used to compute the relationships between users and items by evaluating the user's preference for an item based on the rating of similar items by the same user. The latent factor model transforms both items and users to the same latent factor space and makes them directly comparable [28].

Furthermore, SVD++ exploits explicit feedback, comprising users' input about their interests in items or products, as well as implicit feedback that includes indirect opinions by analyzing user behavior. Guo et al. improved an SVD++-based model to maintain the trust problem in SVD++ via a trust-based recommendation model that integrates explicit and implicit rates with a trust influence [29]. Shi et al. [30] used SVD++ to develop a rating prediction model based on user embedding (UE-SVD++). They exploited the users' potential explicit feedback from the rating data and constructed a user embedding matrix using the proposed user-wise mutual information values in order to enrich the user model further with explicit feedback. Moreover, to improve their model's accuracy, the user-embedding matrix was added to the existing user bias and implicit parameters in the SVD++.

3.3. Nonnegative Matrix Factorization

NMF is a MF method used for analyzing high-dimensional data [9]. NMF automatically extracts features from a set of nonnegative data vectors.

A matrix X is factorized into two matrices P and Q, with the property that all three matrices have no negative elements, as shown in the following equation:

$$X = PQ^T \tag{2}$$

One of the two main differences between NMF and SVD is that the factors of NMF contain only positive values, whereas the factors of SVD and SVD++ accept both positive and negative entries. The other is that SVD produces unique factors, whereas NMF factors are non-unique, making NMF more appropriate for privacy and protection requirements [33].

This algorithm has two primary benefits. First, NMF is considered one of the most typical decomposition tools for extracting important features from a matrix [7]. The second benefit is that it allows the grouping of trajectories based on a common semantic structure [34].

NMF is used extensively in recommendation systems for content- and collaborative filtering-based recommendations. Massimo et al. [34] used NMF in their location-based recommender system by clustering users' POI visit trajectories based on geographical and temporal features. Furthermore, they proposed an NMF-based model for a POI recommender system that incorporates geographical influence and textual data [35]. For this study, we analyzed the use of NMF in a location-based recommendation system by comparing it with other algorithms.

4. Experiments

4.1. Experimental Dataset

The dataset used for training the recommendation system was the Yelp dataset [36]. It is a publicly available dataset of a real-world LBSN. The Yelp dataset contains four types of information: business, review, user, and check-in information. It comprises 5,200,000 user reviews with information on 174,000 businesses in 10 metropolitan areas, namely Montreal, Calgary, Toronto, Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison, and Cleveland.

Each type of data in the dataset is available in the form of separate JSON objects. A business object contains information about the type of business, location, rating, categories, and business name, as well as a unique identifier. A review object includes a rating and reviews text, as well as a unique business ID and user ID. A user object includes the user's friend mapping and the metadata associated with the user. A check-in object includes the check-ins at a business. Table 3 summarizes the information in the Yelp dataset.

Table 3. Yelp Dataset information.

Reviews	Businesses	Metropolitan Areas	Users
8,635,403	160,585	10	2,189,457

In this study, we considered business and review files; the business files contain all places recorded on Yelp along with different categories and information, whereas the review files contain all customers' reviews of each business. Moreover, since we are only interested in restaurants, we filtered the businesses to only places categorized as restaurants and the reviews to those from users who gave 10 or more reviews.

4.2. Evaluation Metrics

In this experiment, two primary error metrics were used to estimate the recommendation accuracy: RMSE and MAE. RMSE and MAE are both used as standard metrics to measure a model's performance. The primary difference between them is the weight of errors: MAE provides the same weight to all errors, whereas RMSE provides additional weight to errors with larger absolute values [37]. The MAE is often used in recommender systems to measure the average absolute deviation between the user's true rating and predicted rating. Many recommender systems have used the MAE for evaluating performance. The following equation was used to compute the MAE [38].

$$MAE = \frac{\sum_{i=1}^{n} |ri - pi|}{n} \tag{3}$$

where ri denotes the actual ratings, pi denotes the predicted ratings, and n denotes the number of ratings.

The RMSE is used to evaluate multiple recommender systems by computing the average value of all absolute value differences between the predicted ratings and the user's true ratings. The RMSE can be computed using the following equation [38]:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (ri - pi)^2} \tag{4}$$

where ri denotes the actual ratings, pi denotes the predicted ratings, and n denotes the number of ratings.

4.3. Experimental Setup

We followed the process described in Figure 1 to analyze the recommender algorithms' efficiency. In Step 1, we selected the Yelp dataset, which has multiple files, as described above. In our experiment, we selected two files, namely business and user reviews. In Step 2, we filtered the business data based on their category and selected only the businesses in the restaurant category. Due to the computational limitations of the large files, we limited the content of the review files to only 359 reviews. Then, we merged the business and review files into one file. Subsequently, we filtered the data to include only unique businesses with one or more reviews. In Step 3, we implemented three recommendation models, namely SVD, SVD++, and NMF.

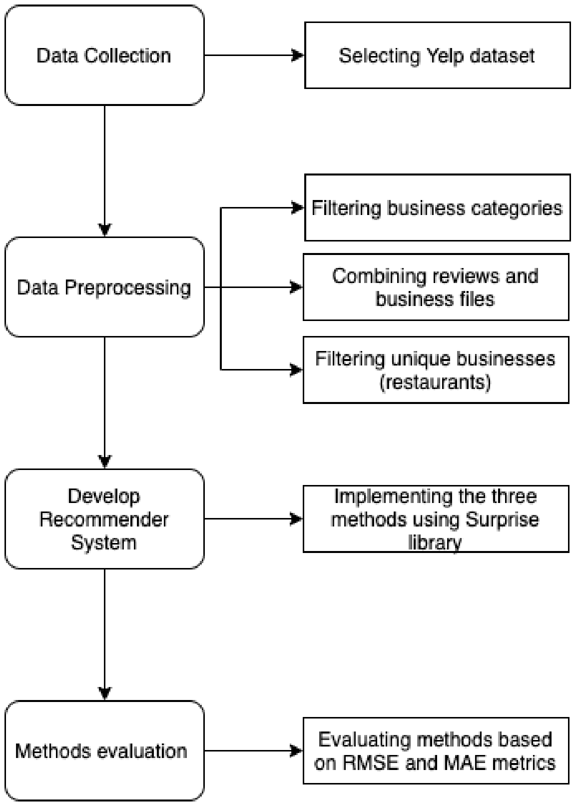

Figure 1. Steps performed to analyze the recommendation methods.

The implementation was performed using the Surprise library [39], an open-source tool offering built-in implementation of different recommendation system methods. Finally, in Step 4, we evaluated the recommender models using the RMSE and MAE metrics. We discuss the evaluation of each method in Section 5.

5. Performance Comparison

The implemented recommender system was based on CF, which predicts the user's rating based on his/her previous ratings. We selected 80% of the dataset as the training set and the remaining 20% as the testing test. The system was applied to the three recommendation methods, which were evaluated using RMSA and MAE, as aforementioned. The testing model was randomly split into five folds.

Table 4 presents a performance comparison of recommendation methods based on the RMSE and MAE.

Table 4. Performance comparison of the recommendation methods, based on the RMSE and MAE metrics.

Root Mean Square Error (RMSE)						
Methods	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Average**
SVD	1.09104711	1.09110819	1.09129044	1.09128233	1.09386881	**1.0913**
SVD++	1.09515548	1.09543396	1.0948695	1.09637749	1.0939797	1.0952
NMF	1.17019635	1.16971371	1.16972419	1.17035167	1.17013915	1.1700
Mean absolute error (MAE)						
Methods	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Average**
SVD	0.85807086	0.8573797	0.85893667	0.85781816	0.86049533	0.8583
SVD++	0.85673553	0.85692388	0.85634049	0.85816445	0.85572834	**0.8568**
NMF	0.90339271	0.90361794	0.9030141	0.90363626	0.90316969	0.9033

The RMSE index of the SVD method demonstrated the lowest error rate, at 1.0913. SVD++ demonstrated an RMSE of 1.0952, and NMF had the largest value, at 1.1700.

The MAE of SVD++ decreased to 0.8568, whereas SVD achieved a value close to SVD++, with an error rate of 0.8583. NMF demonstrated a larger MAE value, with an error rate of 0.9033. Figures 2 and 3 compare the performance of the three methods in graphs.

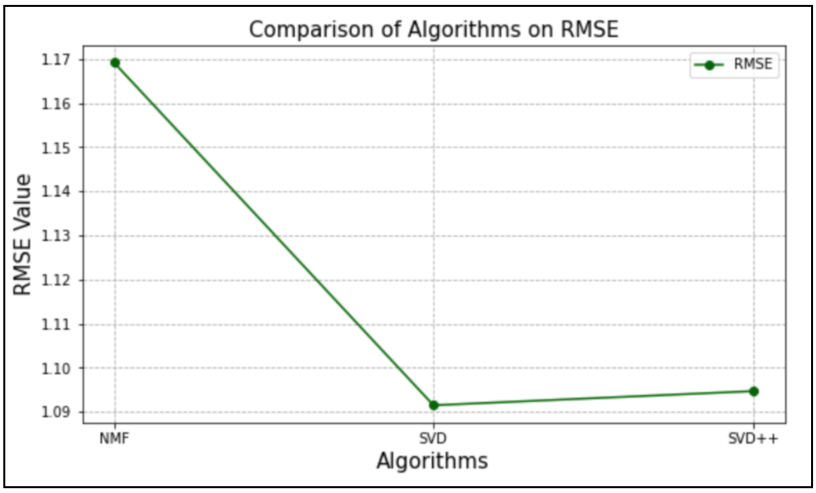

Figure 2. Method comparison based on the RMSE metric.

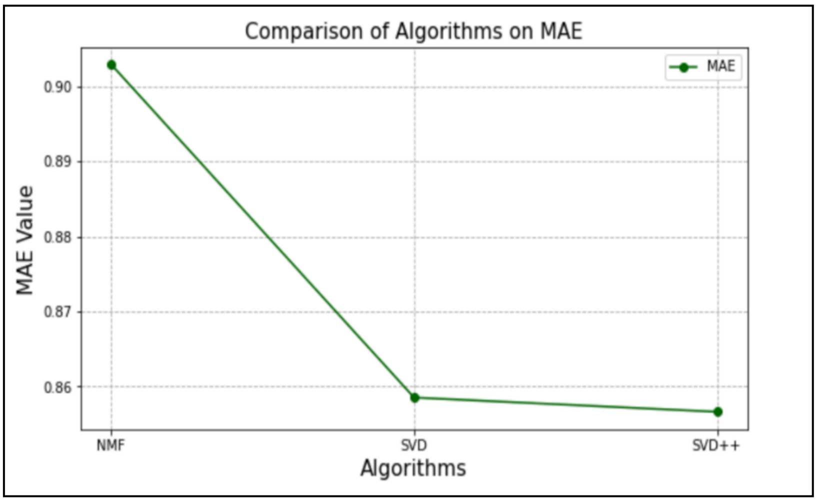

Figure 3. Method comparison based on the MAE metric.

6. Discussion

CF involves analyzing relationships between users and interdependencies among items to identify new user–item associations so that the user discovers new interests based on past behavior. One advantage of CF is that users can gain broader exposure to many suitable restaurants. Another advantage of CF is that CF is domain free—there is no dependence on domain knowledge. Simultaneously, CF faces several main challenges: scalability, sparsity, privacy, and structured recommendations.

These challenges have been addressed by MF. MF-based CF methods can reduce dimensionality by projecting the data matrix into a lower dimension space, so that what appears random and chaotic at high dimensions becomes structured and grouped at lower dimensions. Nevertheless, MF models are sensitive to the amount of information available. MF performs poorly given high levels of sparsity, but the MF models' accuracy improves as the information level increases.

Different MF models have been proposed and used in the literature [21,38,40], such as SVD, probabilistic matrix factorization (PMF), NMF, principal component analysis (PCA), group-specific SVD, and Funk MF. Moreover, hybrid MF and deep-learning MF have been proposed and recently adopted [20,41].

Among these models, SVD and SVD++ models have been proven effective in producing high-quality recommendations and facing the challenges of CF [30,32]. NFM has proven effective in dealing with high-dimensional data due to the nonnegativity constraints [40,41]. Therefore, we aimed to investigate these three methods, specifically their prediction accuracy and storage requirements.

In the Performance Comparison section, the comparison between RMSA and MAE showed that the lowest RMSA was achieved with SVD, followed by SDV++ and NMF. The MAE values were the lowest for SDV++, followed by SDV and NMF.

The results show that using SVD seems to be efficient for collaborative filtering to predict restaurant ratings. The RMSE obtained via the SVD approach was lower than those obtained by SDV++ and NMF were, indicating that the SVD approach can provide more accurate predictions of ratings than the other methods can. SVD is a dimensionality-reducing method capable of reducing the amount of storage required to store the ratings. The combination of reduced data-storage requirements and improved accuracy is a compelling justification for using SVD methods on the subject dataset.

7. Conclusions

In this study, we presented a comparative study of three MF algorithms, namely SVD, SVD++, and NMF. The analysis of these algorithms was based on LBSNs. The primary task of the implemented recommender system is to predict each user's restaurant ratings and make recommendations based on these predictions. This experiment was evaluated using two performance metrics, namely RMSE and MAE. The SVD method confirmed its efficiency by having the lowest RMSE, whereas the SVD++ method exhibited the lowest MAE metric.

Future directions for our work include using a large dataset to fine-tune the model and improve its performance. In addition, alternative MF approaches can be explored, such as hybrid MF, which combines both content and collaborative data to arrive at more relevant and accurate recommendations. Moreover, deep-learning MF could be explored to make more accurate predictions.

Author Contributions: N.A. conceived, designed, and performed the experiment; analyzed and interpreted the data; and drafted the manuscript. A.A.-N. co-supervised the analysis, reviewed and edited the manuscript, and contributed to the discussion. H.A. reviewed the manuscript and supervised this study. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://www.yelp.com/dataset (accessed on 20 November 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, C.; Bai, L.; Zhang, C.; Yuan, Q.; Han, J. Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 1245–1254.
2. Cui, Y.; Deng, L.; Zhao, Y.; Yao, B.; Zheng, V.W.; Zheng, K. Hidden POI Ranking with Spatial Crowdsourcing. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 814–824.
3. Ye, M.; Yin, P.; Lee, W.-C.; Lee, D.-L. Exploiting Geographical Influence for Collaborative Point-of-Interest Recommendation. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information—SIGIR '11, Beijing, China, 24–28 July 2011; pp. 325–334.
4. Zhang, H.; Gan, M.; Sun, X. Incorporating Memory-Based Preferences and Point-of-Interest Stickiness into Recommendations in Location-Based Social Networks. *ISPRS Int. J. Geo Inf* **2021**, *10*, 36. [CrossRef]
5. Ference, G.; Ye, M.; Lee, W.-C. Location Recommendation for Out-of-Town Users in Location-Based Social Networks. In Proceedings of the 22nd ACM International Conference on Information & Knowledge Management—CIKM '13, San Francisco, CA, USA, 27 October–1 November 2013; pp. 721–726.
6. Koren, Y.; Bell, R.; Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
7. Bobadilla, J.; Bojorque, R.; Esteban, A.H.; Hurtado, R. Recommender Systems Clustering Using Bayesian Non Negative Matrix Factorization. *IEEE Access* **2017**, *6*, 3549–3564. [CrossRef]
8. Paatero, P.; Tapper, U. Positive Matrix Factorization: A Non-Negative Factor Model with Optimal Utilization of Error Estimates of Data Values. *Environmetrics* **1994**, *5*, 111–126. [CrossRef]
9. Seung, D.; Lee, L. Algorithms for Non-Negative Matrix Factorization. *Adv. Neural Inf. Process. Syst.* **2001**, *13*, 556–562.
10. Facebook. *Facebook Reports Fourth Quarter and Full Year 2020 Results*; Facebook Inc.: Menlo Park, CA, USA, 2020.
11. Symeonidis, P.; Ntempos, D.; Manolopoulos, Y. Location-Based Social Networks. In *Recommender Systems for Location-Based Social Networks*; Springer: New York, NY, USA, 2014; pp. 35–48.
12. Ma, X.; Zhu, J.; Zhang, S.; Zhong, Y. Multi-factor Fusion POI Recommendation Model. In Proceedings of the International Conference of Pioneering Computer Scientists, Engineers and Educators, Taiyuan, China, 18–21 September 2020; pp. 21–35.
13. Wang, M.; Lu, Y.; Huang, J. SPENT: A Successive POI Recommendation Method Using Similarity-Based POI Em-Bedding and Recurrent Neural Network with Temporal Influence. In Proceedings of the IEEE International Conference on Big Data and Smart Computing (BigComp), Kyoto, Japan, 27 February–2 March 2019; pp. 1–8.
14. Zhao, S.; Zhao, T.; Yang, M.R.L.; King, I. STELLAR: Spatial-Temporal Latent Ranking for Successive Point-of-Interest Recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
15. Zhao, P. Where to Go Next: A Spatio-Temporal Gated Network for Next POI Recommendation. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 2512–2524. [CrossRef]
16. Qian, T.; Liu, B.; Nguyen, Q.V.H.; Yin, H. Spatiotemporal Representation Learning for Translation-Based POI Recommendation. *ACM Trans. Inf. Syst* **2019**, *37*, 1–24. [CrossRef]
17. Cai, L.; Xu, J.; Liu, J.; Pei, T. Integrating spatial and temporal contexts into a factorization model for POI Recommendation. *Int. J. Geogr. Inf. Sci* **2018**, *32*, 524–546. [CrossRef]
18. Yin, C.; Shi, L.; Sun, R.; Wang, J. Improved Collaborative Filtering Recommendation Algorithm Based on Differential Privacy Protection. *J. Supercomput.* **2020**, *76*, 5161–5174. [CrossRef]
19. Jia, Z.; Yang, Y.; Gao, W.; Chen, X. User-Based Collaborative Filtering for Tourist Attraction Recommendations. In Proceedings of the 2015 IEEE International Conference on Computational Intelligence & Communication Technology, Ghaziabad, India, 13–14 February 2015; pp. 22–25.
20. Xue, F.; He, X.; Wang, X.; Xu, J.; Liu, K.; Hong, R. Deep Item-Based Collaborative Filtering for Top-N Recommendation. *ACM Trans. Inf. Syst. (TOIS)* **2019**, *37*, 1–25. [CrossRef]
21. Pujahari, A.; Sisodia, D.S. Model-Based Collaborative Filtering for Recommender Systems: An Empirical Survey. In Proceedings of the 1st International Conference on Power, Control and Computing Technologies (ICPC2T), Raipur, India, 3–5 January 2020; pp. 443–447.
22. Jiang, S.; Qian, X.; Shen, J.; Fu, Y.; Mei, T. Author Topic Model Based Collaborative Filtering for Personalized POI Recommendation. *IEEE Trans. Multimed.* **2015**, *17*, 907–918. [CrossRef]
23. Berry, M.W.; Dumais, S.T.; O'Brien, G.W. Using linear algebra for intelligent information retrieval. *SIAM Rev.* **1995**, *37*, 573–595. [CrossRef]
24. Yuan, X.; Han, L.; Qian, S.; Xu, G.; Yan, H. Singular value decomposition based recommendation using imputed data. *Knowl. Based Syst.* **2019**, *163*, 485–494. [CrossRef]

25. Chai, Z.; Li, Y.-L.; Han, Y.-M.; Zhu, S.-F. Recommendation System Based on Singular Value Decomposition and Multi-Objective Immune Optimization. *IEEE Access* **2018**, *7*, 6060–6071. [CrossRef]
26. Bakir, C. Collaborative Filtering with Temporal Dynamics with Using Singular Value Decomposition. *Teh. Vjesn. Tech. Gaz.* **2018**, *25*, 130–135. [CrossRef]
27. Chen, H.; Arefin, M.S.; Chen, Z.; Morimoto, Y. Place Recommendation Based on Users Check-in History for Location-Based Services. *Int. J. Netw. Comput.* **2013**, *3*, 228–243. [CrossRef]
28. Koren, Y. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 426–434.
29. Guo, G.; Zhang, J.; Yorke-Smith, N. A Novel Recommendation Model Regularized with User Trust and Item Ratings. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 1607–1620. [CrossRef]
30. Shi, W.; Wang, L.; Qin, J. User Embedding for Rating Prediction in SVD++-Based Collaborative Filtering. *Symmetry* **2020**, *12*, 121. [CrossRef]
31. Junior, A.F.; Medeiros, F.; Calado, I. An Evaluation of Recommendation Algorithms for Tourist Attractions. In Proceedings of the Thirty Second International Conference on Software Engineering and Knowledge Engineering (SEKE 2020), Pittsburgh, PA, USA, 9–11 July 2020.
32. Hu, Z. SSL-SVD: Semi-supervised Learning–based Sparse Trust Recommendation. *ACM Trans. Internet Technol.* **2020**, *20*, 1–20. [CrossRef]
33. Athanasiadis, C.; Hortal, E.; Koutsoukos, D.; Lens, C.Z.; Asteriadis, S. Personalized, Affect and Perfor-Mance-Driven Computer-Based Learning. In Proceedings of the 9th International Conference on Computer Supported Education, Porto, Portugal, 21–23 April 2017; pp. 132–139.
34. Massimo, D.; Ricci, F. Clustering Users' POIs Visit Trajectories for Next-POI Recommendation. In *Information and Communication Technologies in Tourism 2019, Proceedings of the International Conference in Nicosia, Cyprus, 30 January–1 February 2019*; Springer: Cham, Switzerland, 2019; pp. 3–14.
35. Liu, B.; Fu, Y.; Yao, Z.; Xiong, H. Learning Geographical Preferences for Point-of-Interest Recommendation. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 1043–1051.
36. Yelp-Dataset. Yelp Open Dataset. Available online: https://www.yelp.com/dataset (accessed on 20 November 2021).
37. Chai, T.; Draxler, R.R. Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)?—Arguments against Avoiding RMSE in the Literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [CrossRef]
38. Ziad, S.; Najafi, S. Evaluating Prediction Accuracy for Collaborative Filtering Algorithms in Recommender Systems. Bachelor's Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2016.
39. Hug, N. Surprise: A Python Library for Recommender Systems. *J. Open Source Softw.* **2020**, *5*, 2174. [CrossRef]
40. Bokde, D.; Sheetal, G.; Debajyoti, M. Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey. *Procedia Comput. Sci.* **2015**, *49*, 136–146. [CrossRef]
41. Rui, C.; Qingyi, H.; Yan-Shuo, C.; Bo, W.; Lei, Z.; Xiangjie, K. A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks. *IEEE Access* **2018**, *6*, 64301–64320.