Data Min Knowl Disc (2018) 32:1–24
https://doi.org/10.1007/s10618-017-0516-z

Noise-tolerance matrix completion for location recommendation

Bin Xia[1] · **Tao Li**[2,3] · **Qianmu Li**[1] · **Hong Zhang**[1]

Received: 11 June 2016 / Accepted: 23 May 2017 / Published online: 5 June 2017

Abstract Due to the sharply increasing number of users and venues in Location-Based Social Networks, it becomes a big challenge to provide recommendations which match users' preferences. Furthermore, the sparse data and skew distribution (i.e., structural noise) also worsen the coverage and accuracy of recommendations. This problem is prevalent in traditional recommender methods since they assume that the collected data truly reflect users' preferences. To overcome the limitation of current recommenders, it is imperative to explore an effective strategy, which can accurately provide recommendations while tolerating the structural noise. However, few study concentrates on the process of noisy data in the recommender system, even recent matrix-completion algorithms. In this paper, we cast the location recommendation as a mathematical matrix-completion problem and propose a robust algorithm named Linearized Breg-

Responsible editor: G. Karypis.

✉ Bin Xia
ben.binxia@gmail.com

✉ Tao Li
taoli@cs.fiu.edu

Qianmu Li
liqianmu@126.com

Hong Zhang
zhhong@njust.edu.cn

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

[2] School of Computing and Information Sciences, Florida International University, Miami, FL, USA

[3] School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China

man Iteration for Matrix Completion (LBIMC), which can effectively recover the user-location matrix considering structural noise and provide recommendations based solely on check-in records. Our experiments are conducted by an amount of check-in data from Foursquare, and the results demonstrate the effectiveness of LBIMC.

Keywords LBSNs · Recommender system · Check-in · Structural noise · Bregman iteration

1 Introduction

Location recommendation has attracted a lot of research attention with the rapid development of Location-Based Social Networks (LBSN). For instance, Foursquare, which is the famous LBSN established in 2009, has collected more than 8 billion check-in data and contains at least 50 million active users each month worldwide. By using Foursquare APP (i.e., Swarm), users can post check-in records while visiting the locations, and share the experience with friends. However, the sharply increasing number of users and locations brings a big challenge to capture users' preferences and provide appropriate recommendations.

Recommender systems shed some light on the solution to overcome the problem of information overload, as they excavate valuable information to match users' preferences in such big data (Zhou et al. 2015a, b). Nonetheless, similar to classic item recommendation platforms (e.g., Amazon and Netflix), in LBSNs, there may have many fake check-in records created by those who intend to manipulate the popularity of target locations (e.g., retail stores and restaurants). In this paper, we consider these fake check-in records as the structural noise. Different from the Gaussian noise, the structural noise often randomly emerges in some specific rows or columns without following any distribution or showing any pattern. For example, in the user-location matrix constructed by check-in records (Fig. 1a), the distribution of check-ins is extremely imbalanced in Location n; Similarly, the check-in records posted by user m are mainly at location 2. For illustration purpose, Fig. 1b presents the check-in records of two users (i.e., user_id 36543482 and user_id 338901468) in our dataset collected from Foursquare (detailed description of the dataset can be found in Sect. 5.1.1). As shown in Fig. 1b, the check-in time of these users (i.e., user 36543482 at Time Square and user 338901468 at New York Hilton Midtown) is continuous and periodic. It is difficult to create such check-in records manually without automatic mechanisms. As a result, these check-ins can be viewed as the simple structural noise which generates the skew distribution. Furthermore, these records account for about 90% of the corresponding users' check-in history and about 3–50% of all check-in records at the corresponding locations. These data, which are defined as structural noise, may not truly reflect users' preference and will affect the effectiveness of recommenders.

The aforementioned simple structural noise can be easily eliminated using simple rules such as the fixed time interval, however, the complex structural noise which does not show any pattern is often difficult for simple filters to identify. For example, Fig. 1c is an example of complex structural noise. It shows the check-in records of a user (i.e., user_id 214418658) in the top 2 locations she visited, where 132 (7.1%) records in

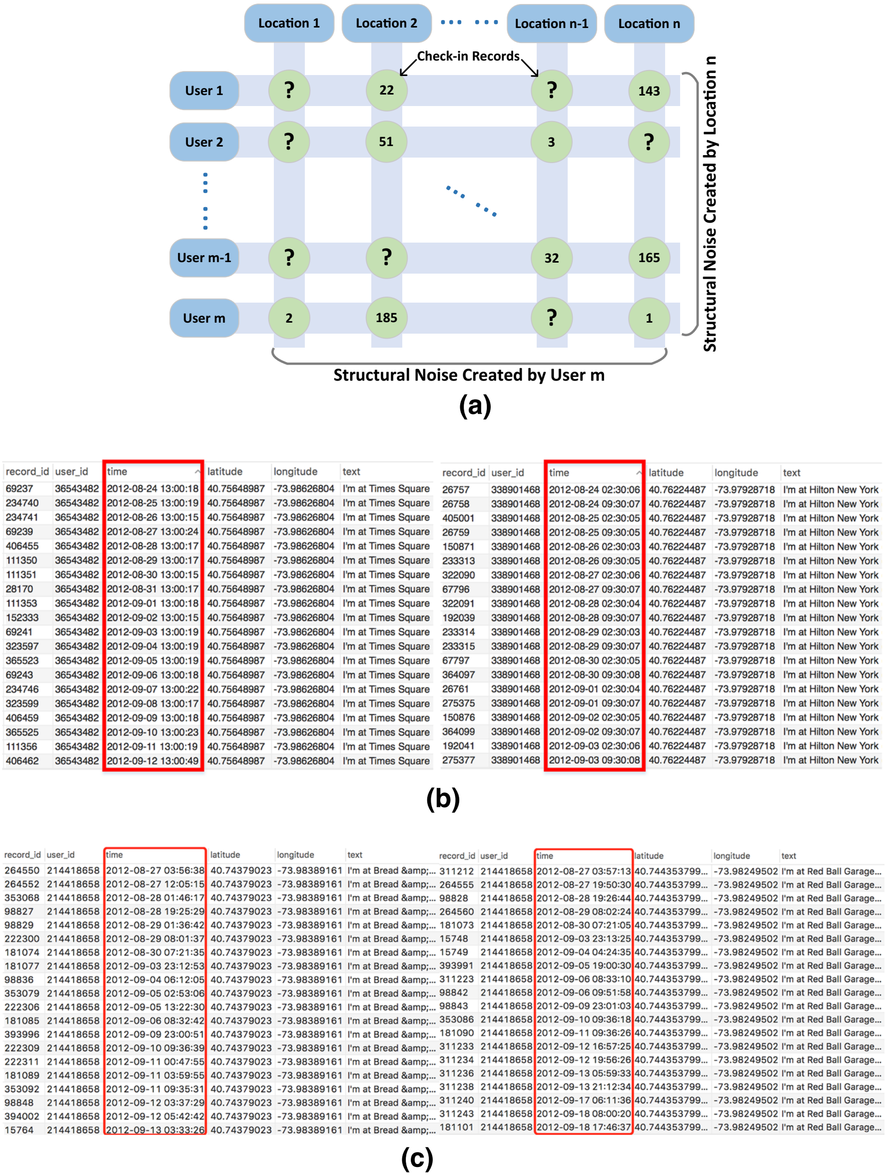

Fig. 1 Structural noise. **a** User-location matrix with structural noise. **b** Examples of simple structural noise in our dataset. **c** Examples of complex structural noise in our dataset

Bread and Butter and 121 (6.5%) in Red Ball Garage. The visit time of these two locations are totally random and do not follow any distribution. In order to identify and eliminate these kinds of structural noise, we should design complex rules or filters based on the domain knowledge (e.g., opening hours and geographical factors).

To overcome this limitation, we propose an effective strategy named Linearized Bregman Iteration for Matrix Completion (LBIMC), solely using the check-in data collected from LBSNs. In this scenario, the user-location matrix is conducted by the number of check-in records which represents the times each user has visited the specific location (Fig. 1a), and we cast the location recommendation problem as the mathematical matrix completion. The matrix completion was described as the solution of minimizing the rank function of the matrix. However, the problem of minimizing the rank function is considered as an NP-hard problem due to the characteristics of the rank function (i.e., the discontinuity and non-convexity). Since the rank of a matrix is equivalent to the L_0-*norm* of the singular value vector of a matrix, and the L_0-*norm* can be smoothed into L_1-*norm* based on the theory of compressed sensing (Donoho 2006). Thus, we employ the nuclear norm which is continuous and convex instead of using the rank function. Currently, there are many effective approaches in compressed sensing to solve such kind of matrix completion problem such as those of Cai et al. (2010) and Chen et al. (2014). However, those approaches have not been fully utilized in recommender system. Therefore, based on sparse check-in data with fake records (i.e., the sparse signal with unknown noise information), we tend to recover the user-location check-in matrix using Bregman iteration algorithm. The completed user-location matrix can intuitively describe users' latent preference even if those locations they've never visited. Particularly, in this case, we assume that the number (i.e., estimated check-in records) located in the row (i.e., user) and column (i.e., location) demonstrates user's preference of the specific location.

To the best of our knowledge, this is the first attempt to solve the problem of location recommendation considering structural noise employing the theory of compressed sensing. Our contribution can be summarized as follows:

- We transform the problem of location recommendation into the mathematical matrix completion employing the theory of compressed sensing.
- We recover the user-location check-in matrix considering the structural noise (i.e., fake check-in records) and show good noise tolerance.
- We evaluate our proposed approach using the check-in data collected from Foursquare, compare it with other state-of-the-art recommender strategies, and demonstrate the effectiveness of LBIMC.

The rest of this paper is organized as follows: Sect. 2 describes relevant works; several mathematical preliminaries used in the derivation process of LBIMC are provided in Sect. 3; Sect. 4 presents our proposed approach of matrix completion for location recommendation; The experimental results are discussed and analyzed in Sect. 5; finally, we summarize our work in Sect. 6.

2 Related work

In recent years, the popularity of smartphones accelerates the development of LBSN. Plenty of meaningful information, such as the location information and user geo-trajectory, can be available in LBSNs. In particular, check-in information, which is the simple and accurate location data, can intuitively describe users' historic visiting records. In terms of this novel contextual data, many effective recommender algorithms have been proposed for location recommendation.

Context-based recommendation. Ye et al. (2011) proposed a unified POI recommendation framework which takes user preference, social influence, and geographical influence into consideration. In their work, they emphasized and proved that the geographical proximities of POIs significantly affect users' behavior. Noulas et al. (2012) found that the majority of people chose the locations those they have never visited in the previous 30 days. According to this observation, they proposed a model using the frequency visiting data based on the individual random walk over a user-venue graph. Bao et al. (2012) presented a recommender system considering users' visiting history and local experts (i.e., people who have many check-in data in specific categories) to match users' preference when they visit a new city. Liu et al. (2013, 2015) also developed a geographical probabilistic factor analysis framework considering various factors such as regional popularity for POI recommendation. The proposed approach can capture users' mobility patterns which are imperative for POI recommendations. In addition, the latent factor method can also model the implicit feedback recommendations considering skewed check-in data. Li et al. (2015) empirically found that users' check-in behaviors are always based on personal preference and social friend interests. According to this observation, they proposed a User Interest Probabilistic Matrix Factorization and a Social Friend Probabilistic Matrix Factorization to model personal preference and social friend interests, respectively. In other categories of recommendations, location data (i.e., check-in) also plays crucial roles. Macedo et al. (2015) used location data and RSVP (i.e., a kind of check-in data in Event-Based Social Networks) for event recommendation. They considered several contextual features such as group, content, location, and time as individual recommenders, and employed listwise Coordinate Ascent (Metzler and Croft 2007) to rank recommendations. There are also some novel context-based recommender systems (Baral and Li 2016; Baral et al. 2016). However, in the context-based model, the collection of contextual data is extremely cost-expensive or even unavailable for researchers.

Opinion spam. Besides these context-based approaches, Jindal and Liu (2008) proposed the concept of opinion spam, which is the early research considering fake reviews in the recommender system. This proposed problem attracts a lot of research attention, and many strategies of fake review detection are presented. Lim et al. (2010) proposed a behavioral approach detecting review spammers who game the system to manipulate some target products, where the model used ensemble behavior scoring approaches to ranking candidate spammers. Mukherjee et al. (2011) proposed an effective technique recognizing groups in which spammers write fake reviews in the similar pattern. In this model, they used frequent pattern mining to discover candidate group and ranked them in terms of spam indicator value using RankingSVM. In their following work, Mukherjee et al. (2012) utilized several behavioral models to detect spammer groups where the models are conducted by the collusion phenomenon among relationships between spammers instead of calculating spam indicator value, and the approach is promising and outperforms state-of-the-art baselines. These studies are interesting trails of recognizing noisy reviews in the recommender system, however, few studies concentrate on skew distribution such as rating, thumbs-up, or implicit ones.

Cross-domain techniques. There are also some cross-domain techniques which have great achievement employed in the recommender system. Cheng et al. (2012) proposed an integrated matrix factorization framework considering the geographical influence which is captured by Multi-center Gaussian Model (MGM) using the probability of users' check-in records. Luo et al. (2014) developed an NMF-based (Non-negative Matrix Factorization) Collaborative Filtering model with a single-element-based approach in which each involved feature is individually updated instead of changing the whole feature matrices in the iterative process. Considering matrix completion as the convex relaxation of rank minimization problem, Cai et al. (2010) proposed an efficient algorithm to solve the nuclear norm minimization problem integrating Bregman iteration (Goldstein and Osher 2009) and proximal forward–backward splitting (Combettes and Wajs 2005). Ma et al. (2011) proposed an algorithm named Fixed Point Continuation with Approximate SVD (FPCA) to approximate the high-dimensional matrix with minimum nuclear norm based on fixed point and Bregman iteration. Besides these studies, many approaches of matrix completion (Melville and Sindhwani 2011; Candes and Plan 2010; Candès and Recht 2009) have also been presented, however, they were not employed in the practical recommender system.

3 Preliminaries

Definition 1 (*Matrix norm*—Meyer 2000) Suppose the rank of matrix $X \in R^{m\times n}$ is r. The singular value decomposition is $X = U\Sigma V^T$ in which:

$$\begin{aligned} &U = [u_1, u_2, \ldots, u_r] \in R^{m\times r}, \quad U^T U = I, \\ &V = [v_1, v_2, \ldots, v_r] \in R^{n\times r}, \quad V^T V = I, \\ &\Sigma = diag(\{\sigma_i | 1 \le i \le r\}), \quad \sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_r > 0. \end{aligned}$$

The nuclear norm of X is defined by:

$$\|X\|_* = \sum_{i=1}^{r} |\sigma_i|$$

The Frobenius norm of X is defined by:

$$\|X\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |X_{ij}|^2}$$

The $L_{2,1}$ norm of X is defined by:

$$\|X\|_{2,1} = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} X_{ij}^2 \right)^{1/2}$$

Definition 2 (*Subdifferential and subgradient*—Rockafellar 2015) Suppose $\Psi(X) \in R^{m\times n}$ is a convex function. If $Z \in R^{m\times n}$ and

$$\Psi(X) \ge \Psi(X_0) + \langle Z, X - X_0 \rangle, \quad \forall X \in R^{m\times n},$$

then, Z is the subgradient of $\Psi(X)$ at $X = X_0$, and $\partial\Psi(X_0)$ represents the set of subgradient at $X = X_0$, while it is also the subdifferential of $\Psi(X)$ at $X = X_0$.

Definition 3 (*Singular value thresholding operator*—Cai et al. 2010) Suppose the rank of matrix $X \in R^{m\times n}$ is r and the singular value decomposition is $X = U\Sigma V^T$. If $\tau \geq 0$, then the corresponding singular value thresholding operator is defined by:

$$D_\tau(X) = U S_\tau(\Sigma) V^T,$$

where $S_\tau(\Sigma) = diag(\{\max(0, \sigma_i - \tau)\})$.

Definition 4 (*Matrix space Bregman distance*—Bregman 1967) Suppose $J(X) : R^{m\times n} \to R$ is a convex function, and $X, X^i \in R^{m\times n}$. Then the Bregman distance between X and X^i is defined by:

$$D_J^{P^i}\left(X, X^i\right) = J\left(X\right) - J\left(X^i\right) - \left\langle P^i, X - X^i \right\rangle,$$

where matrix $P^i \in R^{m\times n}$ is a subgradient in subdifferential $\partial J(X^i)$, and $\langle\cdot,\cdot\rangle$ is the matrix inner product.

Definition 5 (*Proximal operator*—Parikh and Boyd 2014) Suppose $\Psi(X) \in R^{m\times n}$ is a convex function. For $\forall V \in R^{m\times n}$, the proximal operator of $\Psi(X)$ is defined by:

$$prox_{\Psi(X)}(V) = \underset{X\in R^{m\times n}}{argmin}\left\{\frac{1}{2}\|X - V\|_F^2 + \Psi(X)\right\}.$$

Theorem 1 (Cai et al. 2010) *If* $\tau > 0$ *and* $X, Z \in R^{m\times n}$*, then:*

$$prox_{\tau\|X\|_*}(Z) = D_\tau(Z).$$

Theorem 2 (Proximal forward–backward splitting—Combettes and Wajs 2005) *Assume* F_1 *and* F_2 *are lower semi-continuous convex functions in matrix space* $R^{m\times n}$*. If* F_2 *is differentiable and satisfies* $\|\triangledown F_2(U) - \triangledown F_2(V)\|_F \leq \beta\|U - V\|_F$ *when* $\exists\beta \in (0, +\infty)$*, then the convex function problem:*

$$\min_{X\in R^{m\times n}} F_1(X) + F_2(X) \tag{1}$$

has several properties described as follows:

1. *If* $\lim\limits_{\|X\|_F\to+\infty} F_1(X) + F_2(X) = +\infty$*, then Eq. 1 has at least one solution;*
2. *If* $F_1 + F_2$ *is strictly convex function, then Eq. 1 has at most one solution;*
3. *If* F_1 *and* F_2 *satisfy condition* 1 *and* 2*, then Eq. 1 has only one solution.*

For arbitrary initialized X^0 *and* $0 < \delta < 2/\beta$*, the unique solution can be found using the coverage of iterative sequence* X^{i+1} *described by:*

$$X^{i+1} = prox_{\delta F_1(X)}\left(X^i - \delta\triangledown F_2(X^i)\right).$$

Theorem 3 *Assume $\tau > 0$, X, X^*, $W \in R^{m\times n}$, and let:*

$$(X^*)^{(i)} = \max\left\{\left\|W^{(i)}\right\|_2 - \tau, 0\right\} \cdot \frac{W^{(i)}}{\left\|W^{(i)}\right\|_2}, i = 1, 2, \ldots, m,$$

then $prox_{\tau\|X\|_{2,1}}(W) = X^*$ where $X^{(i)}$ represents the i_{th} row in matrix X, and $\left\|W^{(i)}\right\|_2$ is the L_2-norm of $W^{(i)}$.

4 Linearized Bregman iteration algorithm for matrix completion

The primary goal of LBRS is to effectively provide each user the most appropriate recommendations considering the contextual information (e.g., check-in history and social relations). In practice, however, due to the problem of data collection (e.g., users' privacy preserving), contextual information is cost-expensive to collect and difficult to integrate from different data sources. Compared with contexts, check-in data is location-based and easily available from social networks (e.g., Twitter). Considering these check-in records, in Location-Based Recommender System (LBRS), we assume that there exist a set of m users $U = \{u_1, u_2, \ldots, u_m\}$ and a set of n locations $L = \{l_1, l_2, \ldots, l_n\}$ where the count of check-in records for users u_i in locations L can be expressed by $C_{u_i} = \{c_{i1}, c_{i2}, \ldots, c_{in}\}$. In terms of C_{u_i}, a high-dimensional and sparse user-location check-in matrix $R^{m\times n}$ can be constructed where $R^{m\times n} = \{(i, j) \in [n] \times [m] : R_{ij} = c_{ij}\}$. If we can accurately estimate and recover all the missing elements in the check-in matrix $R^{m\times n}$, then the completed matrix $R_C^{m\times n}$ can intuitively demonstrate the predicted satisfaction of location l_j where user u_i has never been. According to the exist of fake check-ins, the observed part $R_o^{p\times q} \in R^{m\times n}(p \leq m, q \leq n)$ is still difficult to truly reflect users' preference. Hence, by utilizing the noisy matrix $R_o^{p\times q}$, it is a big challenge to estimate the missing elements of $R^{m\times n}$.

In order to improve the effectiveness of matrix completion, we need to eliminate the noisy information in $R_o^{p\times q}$. Different from other noises, in a recommender system, the misleading information is often irregularly mixing in specific rows (users) or columns (locations). For instance, some merchants collaborate with customers to falsify data (e.g., reviews and check-ins) for improving their reputation. In this paper, we consider these kinds of misleading information as the structural noise since they are row-based and col-based in matrices. Existing strategies of matrix completion (Candes and Plan 2010; Recht 2011; Keshavan et al. 2009) either ignore the noise problem or just assume that the matrix has random Gaussian noise, thus, they will hardly address the structural noise in practical problems. To this end, we employ the regularization of $L_{2,1}$-*norm* to smooth these noisy information, where the objective function $\min_{X\in R^{m\times n}} \|X\|_* \quad s.t.\ P_\Omega(R) = P_\Omega(X)$ is converted into:

$$\min_{X,Z\in R^{m\times n}} \|X\|_* + \lambda \|Z\|_{2,1} \quad s.t.\ P_\Omega(R) = P_\Omega(X + Z), \tag{2}$$

where Z is the noisy matrix, $\|Z\|_{2,1}$ represents the regularization factor of $L_{2,1}$-*norm* for smoothing noise, and λ is used to weigh the structural noise and the rank of matrix. Because the matrix norm satisfies the triangle inequality and homogeneity, $L_{2,1}$-*norm* regularization can be considered as the constrained convex optimization problem, and Eq. 2 has a global optimal solution. However, it is still difficult to solve Eq. 2, since neither the nuclear norm($\|X\|_*$) nor $L_{2,1}(\lambda\,\|Z\|_{2,1})$ is non-differentiable function.

To overcome this problem, Eq. 2 can be converted into the unconstrained convex optimization as follows:

$$\begin{aligned} \min_{X,Z\in R^{m\times n}} &= J(X,Z) + H(X,Z), \\ s.t.\quad J(X,Z) &= \mu(\|X\|_* + \lambda\,\|Z\|_{2,1}) \\ H(X,Z) &= \frac{1}{2}\,\|P_\Omega(R-X-Z)\|_F^2 \end{aligned} \tag{3}$$

As shown in Eq. 3, it is similar to the L_1-*norm* optimization problem (Bregman 1967) in vector space:

Definition 6 (L_1-*norm optimization problem*) Assume that $J(u)$ and $H(u)$ both are convex functions in R^n where $H(u)$ is differentiable and $J(u)$ is non-differentiable. The normal unconstrained optimization problem is described as:

$$\min_{u\in R^n} J(u) + \mu H(u). \tag{4}$$

To effectively solve this problem, we employ the alternating minimization method and split Bregman iteration algorithm (Goldstein and Osher 2009). The solution of Eq. 3 is described as follows:

Algorithm 1 Alternating Linearized Split Bregman Iteration

Input:
$P_\Omega(R)$: the orthogonal projection operator of observed matrix $R^{m\times n}$;
N: the number of iterations.
Output:
$X^{optimal}$, $Z^{optimal}$.
1: Initialize $Z^0=0$, $P_X^0=0$, $P_Z^0=0$;
2: **for** $k=0$ to N **do**
3: $\quad X^{i+1} = \underset{X\in R^{m\times n}}{argmin}\ \mu\,\|X\|_* - \mu\left\langle P_X^i, X\right\rangle + \frac{1}{2}\left\|P_\Omega(R-X-Z^i)\right\|_F^2$
4: $\quad Z^{i+1} = \underset{Z\in R^{m\times n}}{argmin}\ \mu\lambda\,\|Z\|_{2,1} - \mu\lambda\left\langle P_Z^i, Z\right\rangle + \frac{1}{2}\left\|P_\Omega(R-X^{i+1}-Z)\right\|_F^2$
5: $\quad P_X^{i+1} = P_X^i + \frac{1}{\mu}P_\Omega(R-X^{i+1}-Z^{i+1})$
6: $\quad P_Z^{i+1} = P_Z^i + \frac{1}{\mu\lambda}P_\Omega(R-X^{i+1}-Z^{i+1})$
7: **end for**
8: **return** $X^{optimal} \leftarrow X^{N+1}$, $Z^{optimal} \leftarrow Z^{N+1}$.

Since both $\|X\|_*$ and $\|Z\|_{2,1}$ are non-differentiable, it is difficult to calculate X^{i+1} and Z^{i+1} directly. Fortunately, Bregman (1967) has effectively solved Eq. 4 via defining and using Bregman divergence to represent the non-differentiable function in vector space. With the properties (e.g., convexity and linearity) of Bregman divergence, the minimum of non-differentiable functions $\|X\|_*$ and $\|Z\|_{2,1}$ is easily available.

Since Eq. 4 is the convex optimization problem in vector space, while Eq. 3 is in matrix space. Inspired by Cai et al. (2010), we redefine the equation (Definition 4), and the Bregman divergence between (X, Z) and (X^i, Z^i) in $J(X, Z)$ is represented by:

$$\begin{aligned} D_J^{P^i}(X, Z, X^i, Z^i) &= J(X, Z) - J(X^i, Z^i) \\ &\quad - \left\langle \mu P_X^i, X - X^i \right\rangle - \left\langle \mu\lambda P_Z^i, Z - Z^i \right\rangle, \end{aligned} \tag{5}$$

where $P_X^i \in \partial \left\|X^i\right\|_*$, $P_Z^i \in \partial \left\|Z^i\right\|_{2,1}$, $P^i = (P_X^i, P_Z^i)$. Thus, the iterative solution of Eq. 3 is defined as:

$$\begin{aligned} (X^{i+1}, Z^{i+1}) &= \underset{X,Z\in R^{m\times n}}{argmin} \; D_J^{P^i}(X, Z, X^i, Z^i + H(X, Z)) \\ 0 &\in \partial(D_J^{P^i}(X, Z, X^i, Z^i) + H(H, Z))|_{X^{i+1}} \\ 0 &\in \partial(D_J^{P^i}(X, Z, X^i, Z^i) + H(H, Z))|_{Z^{i+1}} \end{aligned}$$

Given the definition of Bregman divergence in matrix space and Theorems 1– 3, X^{i+1} and Z^{i+1} in Algorithm 1 can be respectively calculated as follows:

- The solution of X^{i+1}:

  In terms of the Property 3 of Theorem 2, we set the iteration number of proximal forward–backward splitting (PFBS) as 1, then:

$$X^{i+1} = prox_{\mu\delta\|X\|_* - \mu\delta\langle P_X^i, X\rangle}\left(X^i + \delta P_\Omega(R - X^i - Z^i)\right), \tag{6}$$

$$P_X^{i+1} = P_X^i - \frac{1}{\mu\delta}\left(X^{i+1} - X^i - \delta P_\Omega(R - X^i - Z^i)\right). \tag{7}$$

  Eq. 6 can be simplified as:

$$X^{i+1} = \underset{Z\in R^{m\times n}}{argmin}\, \mu\delta \left\|X\right\|_* + \frac{1}{2}\left\|X - X^i - \mu\delta P_X^i - \delta P_\Omega(R - X^i - Z^i)\right\|_F^2. \tag{8}$$

  Due to $P_X^0 = 0$, Eq. 7 can be simplified as:

$$P_X^{i+1} = -\frac{1}{\mu\delta}X^{i+1} + \frac{1}{\mu}\sum_{i=0}^{i} P_\Omega(R - X^i - Z^i). \tag{9}$$

  Furthermore, assume that:

$$V^i = \delta\sum_{i=0}^{i} P_\Omega(R - X^i - Z^i), \tag{10}$$

then, obviously Eq. 10 can be rewritten as:

$$V^i = V^{i-1} + \delta P_\Omega(R - X^i - Z^i). \tag{11}$$

If we combine Eq. 9 with Eq. 10, then Eq. 9 is simplified into:

$$\mu\delta P_X^i + X^i = V^{i-1}. \tag{12}$$

Combined with Eq. 11 and 12, Eq. 8 can be simplified as:

$$X^{i+1} = \underset{X \in R^{m\times n}}{argmin}\, \mu\delta \|X\|_* + \frac{1}{2}\left\|X - V^i\right\|_F^2. \tag{13}$$

As observed in Eq. 13, we can infer that X^{i+1} is equal to $D_{\mu\delta}(V^i)$ in terms of Theorem 1. Thus, X^{i+1} can be calculated using the iteration described as follows:

$$\begin{cases} V^i & = V^{i-1} + \delta P_\Omega(R - X^i - Z^i) \\ X^{i+1} & = D_{\mu\delta}(V^i) \end{cases}$$

– The solution of Z^{i+1}

Based on the solution of X^{i+1}, similar results of Z^{i+1} and P_Z^{i+1} can be described as:

$$Z^{i+1} = \underset{Z \in R^{m\times n}}{argmin}\, \mu\lambda\delta \|Z\|_{2,1} + \frac{1}{2}\left\|Z - Z^i - \mu\lambda\delta P_Z^i - \delta P_\Omega(R - X^{i+1} - Z^i)\right\|_F^2 \tag{14}$$

$$P_Z^{i+1} = -\frac{1}{\mu\lambda\delta} Z^{i+1} + \frac{1}{\mu\lambda}\sum_{i=0}^{i} P_\Omega(R - X^{i+1} - Z^i) \tag{15}$$

The same Eq. 10, we can also assume that:

$$U^i = \delta \sum_{i=0}^{i} P_\Omega(R - X^{i+1} - Z^i), \tag{16}$$

$$U^i = U^{i-1} + \delta P_\Omega(R - X^{i+1} - Z^i). \tag{17}$$

Combine Eq. 15 with Eqs. 15 and 16 can be simplified as:

$$\mu\lambda\delta P_Z^i + Z^i = U^{i-1}. \tag{18}$$

Equation 14 is updated combining Eqs. 17 and 18:

$$Z^{i+1} = \underset{Z \in R^{m\times n}}{argmin}\, \mu\lambda\delta \|Z\|_{2,1} + \frac{1}{2}\left\|Z - U^i\right\|_F^2. \tag{19}$$

In terms of Theorem 3, the analytic expression of Eq. 19 can be described as:

$$(Z^{i+1})^{(i)} = \max\left\langle \left\|(U^i)^{(i)}\right\|_2 - \mu\lambda\delta, 0\right\rangle \cdot \frac{(U^i)^{(i)}}{\left\|(U^i)^{(i)}\right\|_2}, \quad i = 1, 2, \ldots, m.$$

Thus, we can calculate Z^{i+1} using the iteration described as follows:

$$\begin{cases} U^i & = U^{i-1} + \delta P_\Omega(R - X^{i+1} - Z^i) \\ (Z^{i+1})^{(i)} & = \max\left\langle \left\|(U^i)^{(i)}\right\|_2 - \mu\lambda\delta, 0\right\rangle \cdot \frac{(U^i)^{(i)}}{\left\|(U^i)^{(i)}\right\|_2}, \quad i = 1, 2, \ldots, m \end{cases}$$

Based on the derivation above, the unconstrained $L_{2,1}$-*norm* regularization problem (Eq. 3) can be effectively solved via X^{i+1} and Z^{i+1}. Integrated the algorithms mentioned above, LBIMC is described in Algorithm 2.

Algorithm 2 Linearized Bregman Iteration for Matrix Completion

Input:
$P_\Omega(R)$: the orthogonal projection operator of observed matrix $R^{m\times n}$;
λ: the parameter balancing the structural noise and rank of matrix;
μ: the parameter balancing $(\|X\|_* + \lambda\,\|Z\|_{2,1})$ and $\frac{1}{2}\,\|P_\Omega(R - X - Z)\|_F^2$;
δ: the convergence parameter;
N: the number of iterations.
Output:
$X^{optimal}$, $Z^{optimal}$.

1: Initialize $X^0 = 0$, $Z^0 = 0$, $V^{-1} = 0$, $U^{-1} = 0$;
2: **for** $i = 0$ to N **do**
3: $\quad V^i = V^{i-1} + \delta P_\Omega(R - X^i - Z^i)$;
4: $\quad X^{i+1} = D_{\mu\delta}(V^i)$;
5: $\quad U^i = U^{i-1} + \delta P_\Omega(R - X^{i+1} - Z^i)$
6: $\quad$ **for** $j = 1$ to m **do**
7: $\quad\quad (Z^{i+1})^{(j)} = \frac{(U^i)^{(j)}}{\|(U^i)^{(j)}\|_2} \cdot \max\{\left\|(U^i)^{(j)}\right\|_2 - \mu\lambda\delta, 0\}$;
8: $\quad$ **end for**
9: **end for**
10: **return** $X^{optimal} \leftarrow X^{N+1}$, $Z^{optimal} \leftarrow Z^{N+1}$.

As observed in Algorithm 2, in each iteration, the matrix X^{i+1} can be maintained low-rank, while Z^{i+1} keeps being sparse. In other words, X^{i+1} can effectively retain the preference of each user and characteristics of each location instead of changing them, and Z^{i+1} would accurately match the noise generated by fake data without perturbing other real ones. On the other hand, based on fixed point theorem (Ran and Reurings 2004), V^i and U^i keep being sparse which can save storage space. In Algorithm 2, each iteration has the process of SVD utilized once for the sparse matrix. However, with the increasing number of iteration and dimension of matrix, the computation of SVD is extremely cost-expensive. To this end, Zhang et al. (2015) proposed a low-rank stochastic proximal gradient descent (SPGD) to solve the nuclear norm regularization problem. As described in Zhang et al. (2015), we can construct a low-rank stochastic gradient $\hat{G}_i = U_i V_i^T$ of $\|X\|_*$ at X^i, and calculate X^{i+1} according to:

$$X^{i+1} = \underset{X \in R^{m\times n}}{argmin} \frac{1}{2} \left\| X - \left(X^i - \eta_i \hat{G}_i \right) \|_F^2 + \eta_i \gamma \|X \right\|_*, \tag{20}$$

where $\eta_i > 0$ presents the step size, and γ is the regularization parameter. Compared with SVD, SPGD is more effective in figuring out X^{i+1} in LBIMC where the convergence has been proved, and the time and space complexities are decreased to $O(m+n)$.

5 Evaluation

In this section, we evaluate the effectiveness of our proposed matrix completion approach based on Bregman iteration for location recommendation. In Sect. 5.1, we briefly describe the experimental setup, including the collected dataset of LBSN, compared baselines, and criteria. In Sect. 5.4, we show and discuss the experimental results.

5.1 Experimental setup

5.1.1 Check-in dataset

Foursquare is a famous LBSN that allows people to post check-in records in Twitter[1] when they visit a specific venue based on the location information. By using APIs of Twitter[2] and Foursquare[3], we collected 419,509 tweets published by 49,823 users among 18,899 locations from August 2012 to July 2013 in Manhattan (shown in Fig. 2). This is the dataset utilized in our experiments to evaluate the effectiveness of LBIMC. Figure 3 presents the categories of locations, which are the statistics of our database. In our experiments, we describe the distribution of datasets using sparsity levels which are associated with specific numbers to split the check-in records of users and locations. Figure 4 depicts the distribution of check-in records at each sparsity level for users and locations.

As observed in Fig. 4a, the majority of users belong to sparsity level 1 (i.e., people who have only 1 record) that demonstrates the check-in data is extremely sparse. From Fig. 4b, we can find the similar phenomenon that the minority of locations have the majority of check-in records. Facing this sparse check-in data, traditional methods (e.g., collaborative filtering) hardly address it very well. In order to evaluate our proposed approach, the dataset is split into 12 months and 10 partitions. For the first partition, as shown in Fig. 5, we define the interval of training sets containing 3 months from the timestamp collected the first tweet to T_c and remove the data of locations which are first available in test interval (i.e., the fourth month). For the

[1] This service has been separated from Foursquare and was integrated into Swarm APP in May, 2014.

[2] https://dev.twitter.com/.

[3] https://developer.foursquare.com/.

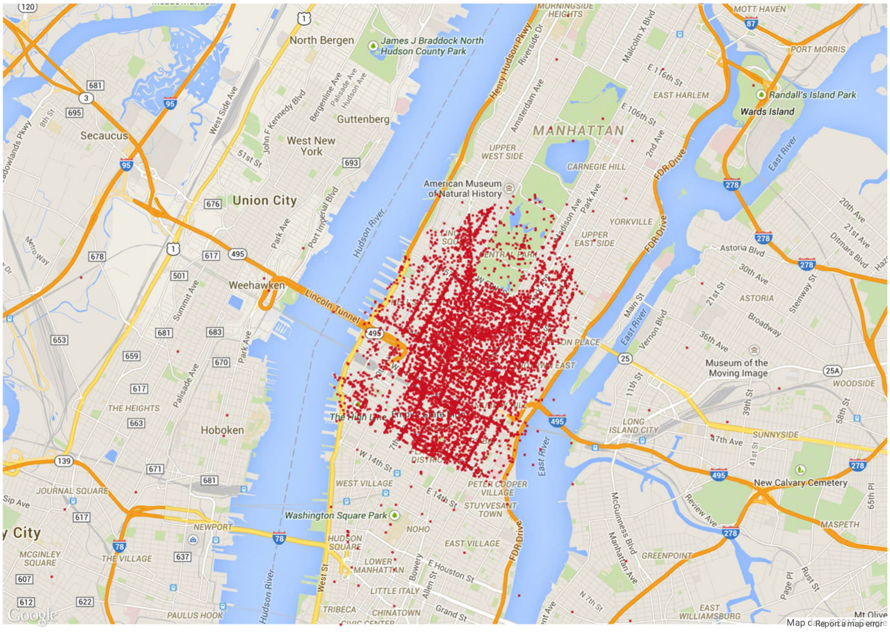

Fig. 2 The check-in locations in Manhattan

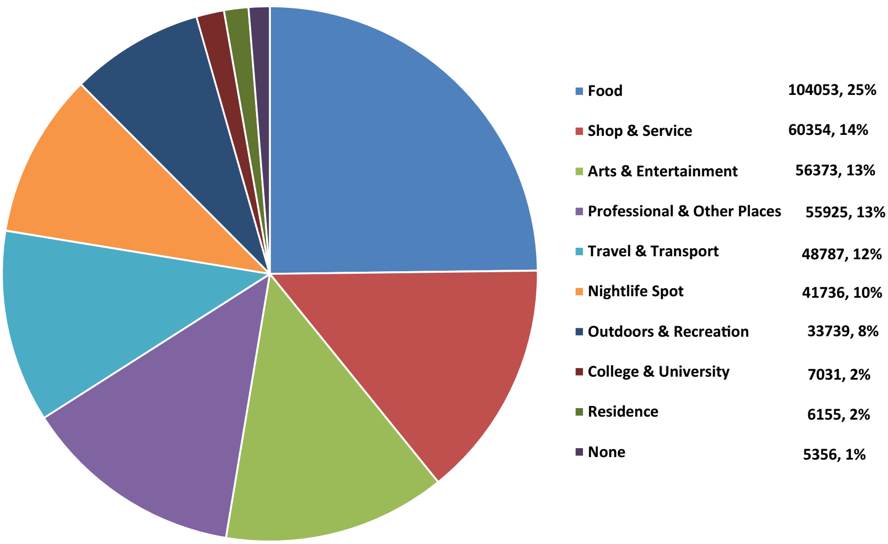

Fig. 3 The check-in popularity among different categories

following partitions, the training set is conducted by datasets (i.e., both training and test) in the last partition, and the interval of the test set is the next month after the training set.

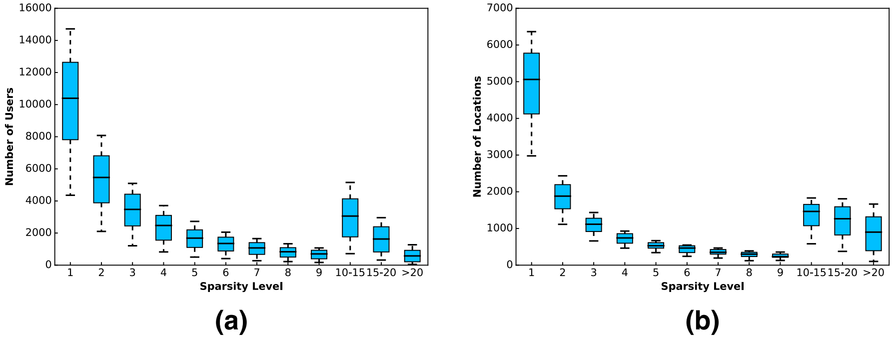

Fig. 4 The distribution of check-in records at each sparsity level for users and locations. **a** Check-in records for users. **b** Check-in records for locations

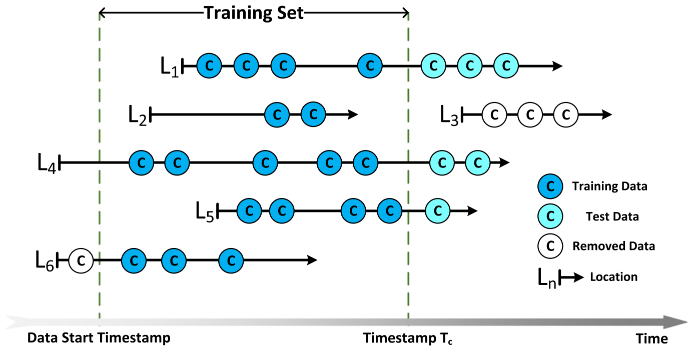

Fig. 5 The partition of training and test datasets

5.2 Baseline venue recommender strategies

To evaluate the effectiveness of our proposed matrix completion method, we compare LBIMC with the following state-of-the-art recommender strategies:

- *UserCF* The assumption of UserCF (User-based Collaborative Filtering) is *people who have similar preferences visit similar venues*. The idea of UserCF is to provide user u_i recommendations in terms of similar users $u_1, u_2, \ldots, u_{i-1}, u_{i+1} \ldots, u_m$ who visit similar locations. UserCF is efficient, however, it is limited in the user Cold-Start problem and the sharply increasing number of users and locations.
- *ItemCF* Different from UserCF which is based on similar users, ItemCF (Item-based Collaborative Filtering) focuses on the similar locations. The assumption of ItemCF is *people will visit venues which have similar characteristics*. ItemCF addresses the User Cold-Start problem and improves the scalability, however, it is limited in the location Cold-Start problem.
- *MP* Besides Collaborative Filtering strategies, recommending the most popular (MP) locations is also a simple and efficient method in the location recommendation. The assumption of MP is *the venue is not bad if many other people have visited*. Obviously, MP still has the location Cold-Start problem.

- *NMF* (Non-negative Matrix Factorization—Lin 2007) is an algorithm in linear algebra where the matrix V^{m*n} is factorized into two matrices W^{m*k} and H^{i*n} with $k < min(m, n)$. In this paper, V^{m*n} represents the check-in matrix conducted by m users and n locations in which W^{m*k} and H^{i*n} describe m users and n locations with k factors, respectively. The recommendation score of location j for user i can be estimated by the product of i_{th} row in W^{m*k} and j_{th} column in H^{i*n}.
- *PMF* (Probabilistic Matrix Factorization—Salakhutdinov and Mnih 2011) is a probabilistic model proposed for addressing recommendation problems based on the extremely huge, sparse, and imbalanced movie datasets (e.g., Netflix). It assumes that *people who rate similar movies are more likely to have the similar preferences*.
- *BPR* (Bayesian Personalized Ranking—Rendle et al. 2009) is a well-known algorithm for providing recommendations from implicit feedbacks. A learning approach based on SGD (stochastic gradient descent) is used to optimize the model considering the maximum posterior estimator of recommendation problems. Since LBIMC is a matrix factorization based algorithm, in this paper, we learn a matrix factorization model with BPR and compare it with our proposed method.
- *ESSVM* (Embedded Space ranking SVM—Xia et al. 2017) is a recently developed context-aware recommender approach which exploits location-dependent features including geospatial data and provides personalized recommendation using ranking SVM. This approach assumes that users' preferences are totally diverse according to individuals, time(i.e., workday and weekend, day and night), and locations(i.e., distance from users' previous check-in).

5.3 Metrics

In order to assess our proposed approach, we use well-known criteria for TopN recommendation, such as Recall, Precision, and Coverage. Assume that, U is the set of all users, $R(u)$ is the recommendation list based on the preference of user u in the training dataset, and $T(u)$ is the set of user u's behaviors in the test dataset. Thus, Recall and Precision can be defined as follows:

$$Recall = \frac{\sum_{u\in U} |R(u) \cap T(u)|}{\sum_{u\in U} |T(u)|}, \tag{21}$$

$$Precision = \frac{\sum_{u\in U} |R(u) \cap T(u)|}{\sum_{u\in U} |R(u)|}. \tag{22}$$

Recall and Precision can intuitively describe the effectiveness of strategies, however, another goal of a recommender system is to, as far as possible, equally recommend all items. In this paper, we define Coverage using (1) ratio of the number of recommendations to that of all locations, and (2) Gini Index based on the number of times each location is recommended (Shani and Gunawardana 2011):

$$Coverage = \frac{| \cap_{u\in U} \ R(u)|}{|L|}, \tag{23}$$

where $|L|$ is the number of all locations in the dataset, and:

$$Coverage = \frac{1}{n-1}\sum_{i=1}^{n}(2j-n-1)p(l_i), \tag{24}$$

where n is the number of location in training dataset, l_i is the i_{th} location in recommendation list which is ordered by the number of times location l is chosen, and $p(l_i)$ is the ratio of times l are recommended to the number of users in recommender system. Coverage is 0 when all locations are recommended in the same number of times, and much greater than 0 when some popular location are always chosen.

5.4 Results and discussion

In this section, we aim to evaluate the effectiveness of LBIMC and answer the following questions:

- How effective is LBIMC for location recommendation?
- How robust is LBIMC to structural noise in check-in data?
- How robust is LBIMC to sparsity in check-in data?

5.4.1 Effectiveness

In order to assess the effectiveness of LBIMC, we compare it to four state-of-the-art recommender strategies UserCF, ItemCF, MP, and NMF which have described in Sect. 5.2. Figure 6 illustrates the effectiveness of seven approaches at 20 location recommendations and the box plot of each approach is conducted by every test partition, and Fig. 7 shows the comparison between our proposed method and other baseline approaches in the top-N recommendation.

From Figs. 6 and 7, we can observe that LBIMC outperforms other recommendation algorithms in Precision including the context-based approach ESSVM. In particular, our approach improves Precision upon the strongest baseline (NMF) by 46%, while LBIMC enhances upon NMF by 44% in Recall. As observed in Fig. 6c, our approach nearly recommends all locations in our dataset, and Fig. 6d demonstrates that LBIMC also has a good performance of equally recommending each location. Notice that, ItemCF outperforms our approach in the coverage of recommendation, however, Precision and Recall show the drawbacks of this recommender.

5.4.2 Noise-tolerance

To evaluate the noise-tolerance of LBIMC, we randomly noise the check-in data of users and locations in training sets according to following rules: (1) if we first select user u_i, a location l_{ij} is randomly chosen to add noise; (2) if we first select location l_j, the random number of check-in records will is added to most users; (3) the noise, in our experiments, is randomly sampled in the range which is predefined based on the distribution of check-in data. Figure 8 illustrates the performance of baselines and LBIMC in noise-tolerance for locations recommendation.

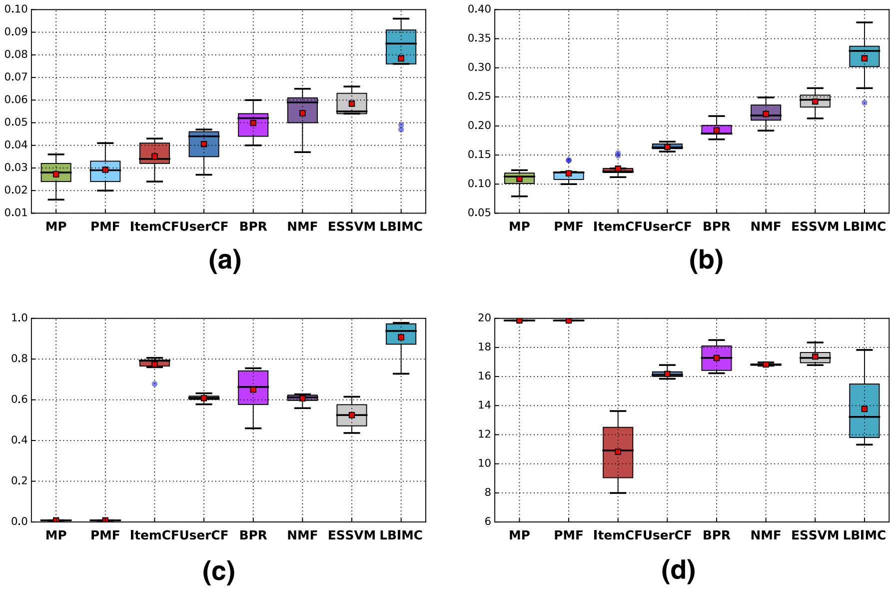

Fig. 6 Location recommendation effectiveness. **a** Precision@20. **b** Recall@20. **c** Coverage_@20. **d** Coverage_Gini@20

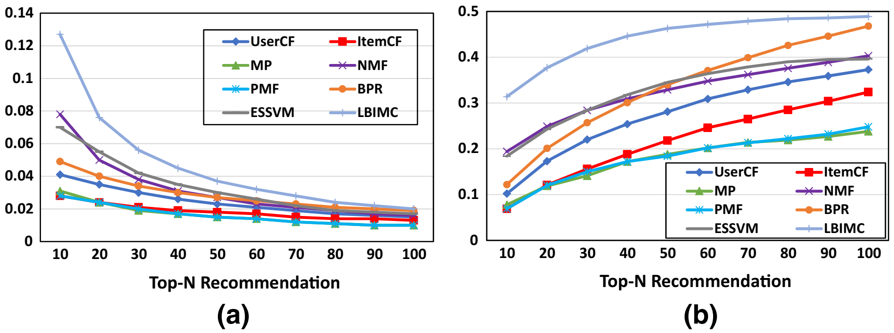

Fig. 7 Performance of TopN location recommendation. **a** Precision@TopN. **b** Recall@TopN

As shown in Fig. 8, our proposed approach outperforms other baselines in terms of Precision and Recall. Due to the effect of randomly added structural noise, the precision value of baseline methods is decreased by 34.2% on average. The performance of UserCF, which provides recommendations based on check-in history of similar people, is greatly impacted since, with structural noise, most users tend to have a similar preference in specific locations. Similarly, structural noise can manipulate the popularity of specific locations, thus, the performance of MP is also degraded. Although ItemCF and NMF have better performance than UserCF and MP, the coverage values of these methods also are sharply decreased due to the structural noise. Compared with baselines which are sensitive to noise, LBIMC shows good performance in noise-tolerance and effectively provides recommendations where the precision value is only

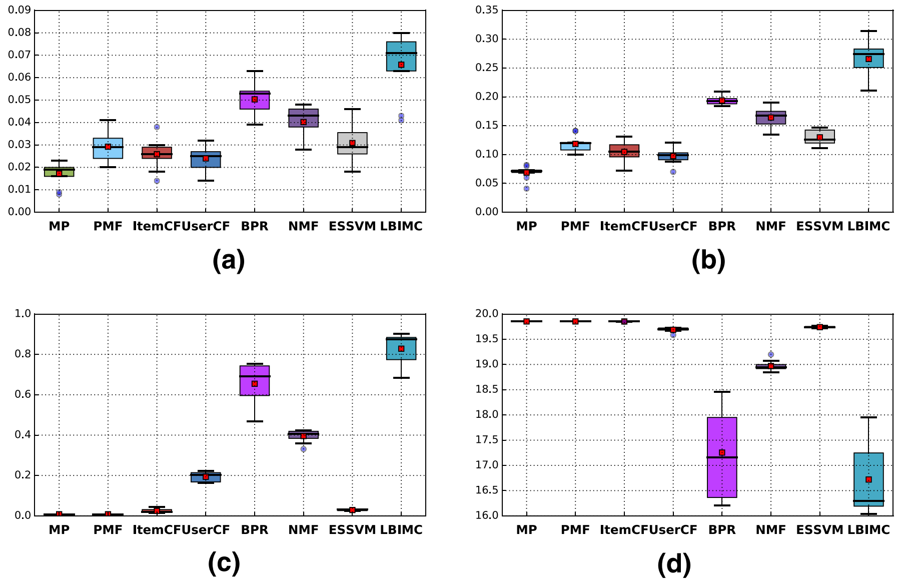

Fig. 8 Location recommendation noise-tolerance **a** Precision@20. **b** Recall@20. **c** Coverage_@20. **d** Coverage_Gini@20

decreased by 15.5%. Notice that, BPR and PMF both have good performance in noise-tolerance, however, the large decrease of Precision and Recall values show drawbacks of these recommenders. Furthermore, the performance of ESSVM decreased significantly with the addition of the structural noise. This shows that the amount of check-ins is more significant than other contexts such as distance and time and the context-aware method ESSVM cannot handle the type of false data. In addition, to further validate the noise-tolerance of our proposed approach, we randomly noise training sets in multi-granularity of users and locations, and the amount of noise in terms of the average number of check-in records for each user (about 8.4 records per user). Figure 9 illustrates the noise-tolerance under different ratios of users and locations for the recommendation.

From Fig. 9, with the increasing ratio of perturbed noise, we observe that LBIMC outperforms other baselines and maintains good performance in noise-tolerance. Furthermore, traditional collaborative filter algorithms (e.g., UserCF and ItemCF) and MP are extremely sensitive to the ratio of noisy data, particularly when the number of recommendations is less than a number of perturbed locations. Figure 10 illustrates the effectiveness of LIBMC in denoising the user-location check-in matrix. In Fig. 10, blue circles represent the original check-in data where the larger radius demonstrates the more records the user has in the specific location, while red ones are the noised records.

As shown in Fig. 10, LIBMC removed most of the noised data from the original check-in dataset. In statistics, LIBMC is able to filter 87.55% noised data which outperforms NMF by 5.23%. On the other hand, as a mathematical matrix completion, LIBMC is also robust for the order of users and locations in check-in matrix. To validate

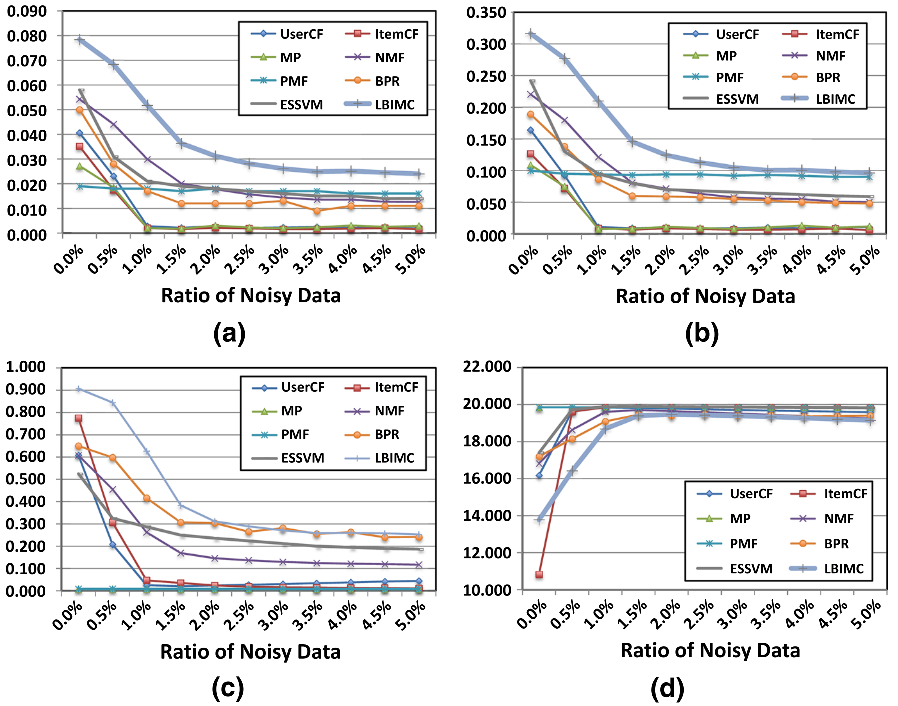

Fig. 9 Location recommendation noise-tolerance. **a** Precision@20. **b** Recall@20. **c** Coverage@20. **d** Coverage_Gini@20

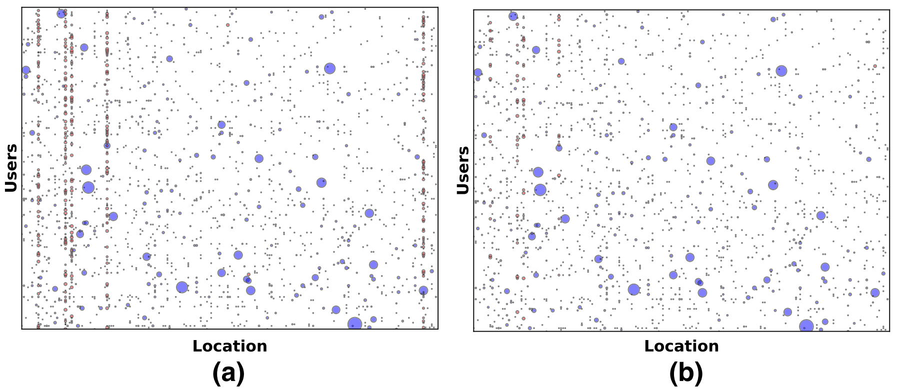

Fig. 10 Example of denoised user-location check-in matrix. **a** Noised check-in matrix. **b** check-in matrix Denoised by LIBMC

this characteristic, LIBMC is compared with NMF and BPR which are the other matrix algorithms in this paper using different matrices comprising of random orders of rows (i.e., users) and columns (i.e., locations). Figure 11 shows the effectiveness and robustness of LIBMC.

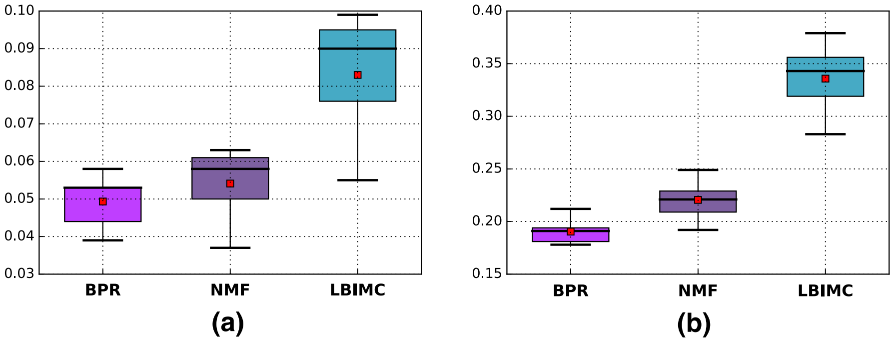

Fig. 11 Performance of LIBMC under random orders in users and locations. **a** Precision@20. **b** Recall@20

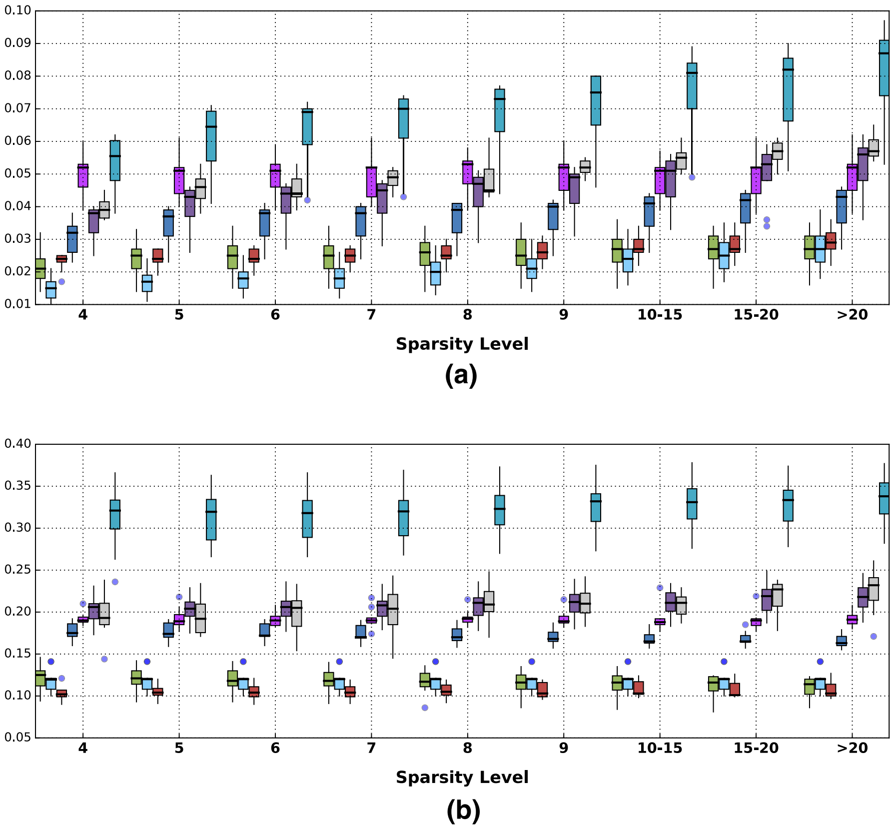

Fig. 12 Location recommendation Sparsity. **a** Precision@20. **b** Recall@20

5.4.3 Sparsity

Another reason for the effective performance of LBIMC is that it is able to address the problem of sparse data (i.e., Cold-Start). To further evaluate the robustness of our approach in sparse check-in data, we compare the effectiveness of LBIMC with aforementioned baselines in ordered sparsity levels synchronously considering the number of user and location. Since users who belong to level 3 or below 3 only have 5 or below 5 check-in records that are unable to validate the algorithms, we aim to evaluate LBIMC in sparsity level 4 (i.e., more than 6 records per user) and above 4. Figure 12 shows the performance of each approach in terms of Precision@20 and Recall@20 for location recommendation.

As observed in Fig. 12, LBIMC has the stronger robustness to sparse check-in data compared with other baseline recommenders. Our proposed approach is able to provide recommendations even in the extreme case (i.e., sparsity level 4) where users have 5–10 check-in records. On the other hand, with the increasing level of sparsity, NMF is the only method that can compete with LBIMC while other baselines are unable to provide effective recommendations based solely on the number of check-in records.

6 Conclusion

In this paper, we cast the location recommendation as a mathematical matrix completion problem and propose an effective approach called LBIMC, which can recover the user-location matrix considering structural noise. In previous works of the recommender system, researchers consider that collected data (e.g., purchase history) truly reflect users' preference. However, some kinds of feedback (e.g., rating, review, and check-in) are always used by merchants to manipulate the popularity of some target locations, since they are freely created. In order to avoid the negative effect of structural noise, LBIMC is designed to overcome the limitations of previous recommenders based on compressed sensing that is used to process sparse and noisy data in the domain of signal process. We employ several popular criteria in recommender system to optimize LBIMC and compare our approach with four state-of-the-art baselines. To the best of our knowledge, this is the first attempt to solve the problem of location recommendation considering structural noise using the theory of compressed sensing.

In the evaluation of an amount of check-in records collected from Foursquare, our proposed approach outperforms other methods and improves upon the strongest baseline Non-negative Matrix Factorization by up to 46% in Precision.

There are several ideas to extend our work in the future. First, for the purpose of personal privacy preserving, users' preferences are often perturbed by noise such as Laplace. Can LBIMC effectively address these noise? Second, in this paper, we only use check-in records to construct user-location matrix ignoring much contextual information such as group, time, and geospatial data. Can we combine these features into user-location matrix? Third, LBIMC is an approach inspired by the research of compressive sensing which cannot explain to users why the locations are recommended. Can we combine LBIMC with other approaches that make recommendations explainable?

Acknowledgements We want to thank Dr. Lei Chen for insightful discussions on matrix completion. We would also want to thank the editor and reviewers for their useful comments and suggestions for improving the paper. The work was supported in part by the National Science Foundation under Grant Nos. IIS-1213026 and CNS-1461926, Chinese National Natural Science Foundation under grant 91646116, and Scientific and Technological Support Project (Society) of Jiangsu Province (No. BE2016776).

References

Bao J, Zheng Y, Mokbel MF (2012) Location-based and preference-aware recommendation using sparse geo-social networking data. In: Proceedings of the 20th international conference on advances in geographic information systems. ACM, pp 199–208
Baral R, Li T (2016) Maps: a multi aspect personalized poi recommender system. In: Proceedings of the 10th ACM conference on recommender systems. ACM, pp 281–284
Baral R, Wang D, Li T, Chen S-C (2016) Geotecs: exploiting geographical, temporal, categorical and social aspects for personalized poi recommendation. In: IEEE 17th international conference on information reuse and integration (IRI), 2016. IEEE, pp 94–101
Bregman LM (1967) The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. USSR Comput Math Math Phys 7(3):200–217
Cai J-F, Candès EJ, Shen Z (2010) A singular value thresholding algorithm for matrix completion. SIAM J Optim 20(4):1956–1982
Candes EJ, Plan Y (2010) Matrix completion with noise. Proc IEEE 98(6):925–936
Candès EJ, Recht B (2009) Exact matrix completion via convex optimization. Found Comput Math 9(6):717–772
Chen L, Yang G, Chen Z, Xiao P, Chen S (2014) Linearized bregman iteration algorithm for matrix completion with structural noise. Chin J Comput 37:1–17
Cheng C, Yang H, King I, Lyu MR (2012) Fused matrix factorization with geographical and social influence in location-based social networks. In: Twenty-sixth AAAI conference on artificial intelligence
Combettes PL, Wajs VR (2005) Signal recovery by proximal forward–backward splitting. Multiscale Model Simul 4(4):1168–1200
Donoho DL (2006) Compressed sensing. IEEE Trans Inf Theory 52(4):1289–1306
Goldstein T, Osher S (2009) The split bregman method for l1-regularized problems. SIAM J Imaging Sci 2(2):323–343
Jindal N, Liu B (2008) Opinion spam and analysis. In: Proceedings of the 2008 international conference on web search and data mining. ACM, pp 219–230
Keshavan R, Montanari A, Oh S (2009) Matrix completion from noisy entries. In: Advances in neural information processing systems, pp 952–960
Li H, Hong R, Zhu S, Ge Y (2015) Point-of-interest recommender systems: a separate-space perspective. In: IEEE international conference on data mining (ICDM), 2015. IEEE, pp 231–240
Lim E-P, Nguyen V-A, Jindal N, Liu B, Lauw HW (2010) Detecting product review spammers using rating behaviors. In: Proceedings of the 19th ACM international conference on information and knowledge management. ACM, pp 939–948
Lin C-J (2007) Projected gradient methods for nonnegative matrix factorization. Neural Comput 19(10):2756–2779
Liu B, Xiong H, Papadimitriou S, Fu Y, Yao Z (2015) A general geographical probabilistic factor model for point of interest recommendation. IEEE Trans Knowl Data Eng 27(5):1167–1179
Liu B, Fu Y, Yao Z, Xiong H (2013) Learning geographical preferences for point-of-interest recommendation. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 1043–1051
Luo X, Zhou M, Xia Y, Zhu Q (2014) An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. IEEE Trans Ind Inf 10(2):1273–1284
Ma S, Goldfarb D, Chen L (2011) Fixed point and Bregman iterative methods for matrix rank minimization. Math Program 128(1–2):321–353
Macedo AQ, Marinho LB, Santos RL (2015) Context-aware event recommendation in event-based social networks. In: Proceedings of the 9th ACM conference on recommender systems. ACM, pp 123–130
Melville P, Sindhwani V (2011) Recommender systems. In: Sammut C, Webb G (eds) Encyclopedia of machine learning. Springer, Berlin, pp 829–838
Metzler D, Croft WB (2007) Linear feature-based models for information retrieval. Inf Retr 10(3):257–274
Meyer CD (2000) Matrix analysis and applied linear algebra, vol 2. Siam, Philadelphia

Mukherjee A, Liu B, Glance N (2012) Spotting fake reviewer groups in consumer reviews. In: Proceedings of the 21st international conference on world wide web. ACM, pp 191–200

Mukherjee A, Liu B, Wang J, Glance N, Jindal N (2011) Detecting group review spam. In: Proceedings of the 20th international conference companion on world wide web. ACM, pp 93–94

Noulas A, Scellato S, Lathia N, Mascolo C (2012) A random walk around the city: New venue recommendation in location-based social networks. In: Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom). IEEE, pp 144–153

Parikh N, Boyd SP (2014) Proximal algorithms. Found Trends Optim 1(3):127–239

Ran AC, Reurings MC (2004) A fixed point theorem in partially ordered sets and some applications to matrix equations. Proc Am Math Soc 132:1435–1443

Recht B (2011) A simpler approach to matrix completion. J Mach Learn Res 12:3413–3430

Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press, pp 452–461

Rockafellar RT (2015) Convex analysis. Princeton University Press, Princeton

Salakhutdinov R, Mnih A (2011) Probabilistic matrix factorization. In: NIPS, vol 20, pp 1–8

Shani G, Gunawardana A (2011) Evaluating recommendation systems. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) Recommender systems handbook. Springer, Berlin, pp 257–297

Xia B, Ni Z, Li T, Li Q, Zhou Q (2017) Vrer: context-based venue recommendation using embedded space ranking SVM in location-based social network. Expert Syst Appl 83:18–29

Ye M, Yin P, Lee W-C, Lee D-L (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, pp 325–334

Zhang L, Yang T, Jin R, Zhou Z-H (2015) Stochastic proximal gradient descent for nuclear norm regularization. arXiv preprint arXiv:1511.01664

Zhou W, Li T, Shwartz L, Grabarnik GY (2015) Recommending ticket resolution using feature adaptation. In: 11th international conference on network and service management (CNSM), 2015. IEEE, pp 15–21

Zhou W, Tang L, Li T, Shwartz L, Grabarnik GY (2015) Resolution recommendation for event tickets in service management. In: 2015 IFIP/IEEE international symposium on integrated network management (IM). IEEE, pp 287–295