Journal of Information Hiding and Privacy Protection
DOI:10.32604/jihpp.2020.010211
Article

Improving POI Recommendation via Non-Convex Regularized Tensor Completion

Ming Zhao* and Tao Liu

School of Computer Science and Engineering, Central South University, Changsha, 410000, China
*Corresponding Author: Ming Zhao. Email: meanzhao@csu.edu.cn
Received: 11 July 2020; Accepted: 21 August 2020

Abstract: The problem of low accuracy of POI (Points of Interest) recommendation in LBSN (Location-Based Social Networks) has not been effectively solved. In this paper, a POI recommendation algorithm based on non-convex regularized tensor completion is proposed. The fourth-order tensor is constructed by using the current location category, the next location category, time and season, the regularizer is added to the objective function of tensor completion to prevent over-fitting and reduce the error of the model. The proximal algorithm is used to solve the objective function, and the adaptive momentum is introduced to improve the efficiency of the solution. The experimental results show that the algorithm can improve recommendation accuracy while reducing the time cost.

Keywords: POI recommendation; tensor completion; proximal algorithm; adaptive momentum

1 Introduction

With the development and popularization of GPS-enabled smartphones, LBSN has derived POI recommendations. At present, the POI recommendation method has the problems of high time cost and low recommendation accuracy. Traditional methods use matrix decomposition and collaborative filtering to recommend [1–2], but the check-in data information is multi-dimensional. Using the matrix will lose some of the information and cannot make full use of the available information. Existing research shows [3], the tensor (a matrix with high dimensional expansion) can better explain multivariate data [4]. Due to the increase of the additional dimension, the prediction performance of the tensor-based method exceeds the matrix-based method [5].

At present, many tensor-based methods have been proposed to improve performance, and Cheng et al. first added consecutive time-stamps to the POI recommendations. They proposed the next personalized POI recommendation problem, established a registration tensor, and utilized the Factorization Personalized Markov Chain (FPMC) [6] to limit motion. To recommend the most likely continuous POI to users, Zhao et al. proposed a space-time potential ranking model that highlights time importance [7]. But the previous approach ignored the personalization factor and failed to combine location information with real-time conditions. Koren et al. [8] proposed a time-aware factorization of personalized Markov chains (TAD-FPMC), in which they use 4-dimensional tensors to store the spatial and temporal characteristics of the checked-in data, and discusses the correlation between two consecutive check-in, and uses time-decay factor to measure the short time interval and long time interval check-in data. Although TAD-FPMC is already advantageous, it still has the following shortcomings. First, computational complexity, using tensor factorization, the method of processing the tensor increases the time for tensor fold and unfold, increases the time-decay factor and makes it difficult to calculate the exact value. Second, the recommendation accuracy is low, the added user dimension is redundant and does not conform to people’s habits, and the convergence of the objective function is slow when the objective function is iterated.

This paper proposes PRNORT (a method of recommending POI through a non-convex regularizer) to reduce time and space cost and improve the accuracy of POI recommendations. Modeled by four-dimensional tensor, which are the current location category, the next location category, time and season, and the regularizer is added to the tensor completion objective function to prevent over-fitting and improve the objective function accuracy. In the process of solving the proximal algorithm, a sparse and low-rank strategy is added to ensure the accuracy, and the momentum is used to accelerate the solution process convergence. The previous momentum is mainly used in the Alternating Direction Method of Multipliers (ADAM) algorithm. A lot of experiments were carried out using realistic data, and compared with the more advanced methods, the conclusion is that our PRNORT method is superior in performance and accuracy.

In short, the contribution of the article can be summarized in the following three points:

I. We model the known information by establishing a fourth-order tensor, in which the user’s dimension is removed and the season dimension is added to reach the long-term preference and short-term preference of the user.

II. Using the tensor completion method and based on the algorithm, a proximal algorithm is added to maintain a “sparse and low-rank” structure, and the expansion and the folding of the tensor are avoided, improve the accuracy of the recommendation.

III. To accelerate the empirical convergence of the solving process of objective function, the adaptive momentum is added, so that a precise approximate result is obtained faster at the time of iteration.

The rest of the organization is organized as follows. In Section 2, we review the work associated with the POI recommendations. The expression of the question and the proposed method are described in detail in Section 3. In Section 4, we evaluate the proposed method and analyze its performance by comparing the actual LBSN data. Finally, we summarized this article in Section 5.

2 Related Works

POI recommendation is a location-based social network (LBSN). After the problem is raised, the collaborative filtering method is widely used to solve this problem. Collaborative filtering is to build the next location matrix using the similarity of the user's characteristics or the similarity of the location [9]. Reference [10] suggests using the collective matrix factorization method to mine interesting locations and activities. Pulling user-related data together to apply collaborative filtering to find like-minded users and similar-patterned activities in different locations, but this method only solves the user's cold-start problem and cannot reasonably recommend a large amount of sparse data, and there is another problem is location’s cold-start, some existing methods utilize user profiles to take advantage of POI-recommended user preferences [11]. Because of the lack of flexibility, it also sacrifices the accuracy of the recommendations.

Recently, many factors and advances have been made to POI recommendations by considering more factors, such as geographic factors, time factors, emotional factors, and social factors, and predicting the next point of interest by combining matrix factorization or other probability models. Reference [12] Power-law distribution is used to simulate the user’s registration behavior, and this geographical influence is included in the POI recommendation. In the literature [13], the method based on probability generation model and the emphasis of matrix factorial factorization are to explore the influence of social factors on POI prediction. Reference [14] used the Gaussian mixture model to predict the POI of the next visit, calculated the probability of the next occurrence in two locations through the direct periodic change of people in the “work place” and “home” state, but this method needs a large number of check-in data as the basis, which cannot solve the problem of sparsity to ensure high accuracy. Lian et al. [15] simulated the neighborhood of geographic locations from the instance level and the regional level. Wang et al. [16] used a weighted matrix factorization framework to combine spatial clustering phenomena. Reference [17] used the symbiotic mode and content of the space-time project to recommend space projects. Ye et al. [18] studied the impact of geographic factors on POI recommendations by combining the geographic location of POI, the geographic sensitivity of POI, and their physical distances. Time is another important factor of POI recommendations. Chang et al. [19] found that people usually go to restaurants around noon and go to the club's periodicity in the evening. Reference [20] studied the influence of joint factors of temporal and spatial information. Reference [21] extended the generalized matrix factorization with temporal dependence by combining geographic factors and time information into matrix decomposition. Yin et al. [22] proposed a sorting-based tensor decomposition framework for subsequent POI recommendations. Liao et al. [23] used heterogeneous semantic, temporal and spatial information to solve real-time or foreign POI recommendations. More importantly, the CF, MF, Markov chain and generation models in these studies are very effective in dealing with the extreme sparsity of POI recommendations, especially when considering space-time integration, but because the factors considered are not complete, lead to low accuracy.

In recent years, to process multi-dimensional information and improve the accuracy of recommendations, tensors (high-dimensional expansion matrices) are widely used for modeling. For matrices, tensors are higher-dimensional matrices, and more information is stored. The obtained recommendation is also more accurate. For the tensor processing, to obtain the estimated value of the sparse data, element filling must be performed in the tensor. In the tensor processing method, there are two categories of tensor factorization and tensor completion. There is only a small difference between tensor factorization and tensor completion. That is, the tensor factorization will have one more message telling the tensor which position is the data we observed, and the rest is the data we have not observed. In a specific method, the tensor factorization does not require estimation of missing information in each iteration. The tensor completion needs to estimate the missing information in each iteration. The specific methods are CP, TD, Tensor-train factorization. Based on these papers, we propose the tensor completion algorithm based on non-convex regularizer. Compared with the previous one, the complexity of time and space is lower. Based on these papers, we propose a tensor completion algorithm based on the non-convex regularizer. Compared with the previous algorithm, the complexity of time and space is lower. add the structure of sparse plus low-rank to ensure the accuracy of iteration. use the adaptive momentum to accelerate the convergence of the objective function.

3 Methodology

3.1 Notation

Tensor, expressed by x, the n-order tensor is represented as $x \in R^{I_1 \times I_2 \dots I_n}$, where $\mathrm{I_n}$ represents the number of elements per dimension. In this article, we need to know some formulas and definitions of tensor, tensor fold is the expansion of a tensor according to a certain dimension, if it is expanded according to the j dimension, can be represented as $x \in R^{I_1 \times I_* / I_1}$. For instance, $x \in R^{\mathrm{I_1 \times I_2 \times \dots I_3}}$ opening according to dimension 1 is a matrix $\mathrm{I_1} \times \mathrm{I_2} \bullet \mathrm{I_3}$,the order reduction treatment is carried out. Tensor unfold is the inverse of the fold process, expressed as $x^{<j>}$,the inner product of the tensor is $\langle \mathrm{X}, Y \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_n=1}^{I_n} X_{i_1 i_{2 \dots in}} Y_{i_1 i_{2 \dots in}}$, the nuclear norm of a tensor is $\| \mathrm{X} \|_{\mathrm{F}} = \sqrt{\langle X, X \rangle}$, kronecker product is,

$$\mathrm{A} \otimes \mathrm{B} = \begin{bmatrix} \mathrm{a_{11}B} & \mathrm{a_{12}B} & \dots & \mathrm{a_{1n}B} \\ \mathrm{a_{21}B} & \mathrm{a_{22}B} & \dots & \mathrm{a_{2n}B} \\ \dots & \dots & \dots & \dots \\ \mathrm{a_{m1}B} & \mathrm{a_{m2}B} & \dots & \mathrm{a_{mn}B} \end{bmatrix} \tag{1}$$

khatri-Rao product: Is a match on a Kronecker product column element:

$$\mathrm{A \Theta B} = [\, \mathrm{a_1 \otimes b_1} \quad \mathrm{a_2 \otimes b_2 \dots a_{\mathit{k}} \otimes b_{\mathit{k}}} \,] \tag{2}$$

The objective function of tensor decomposition is $\min_{\mathrm{A}^{(1)} \dots \mathrm{A}^{(n)}, X} \frac{1}{2} \| X - (\mathrm{A}^{(1)}, \mathrm{A}^{(2)}, \dots \mathrm{A}^{(n)}) \|_F^2$, tensor completion uses a matrix to observe the lack of data, and the objective function is

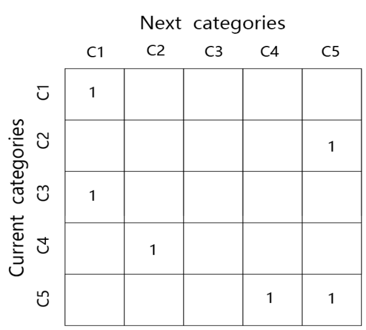
$$\min_{A_{(1)}\ldots A_{(n)},X} \frac{1}{2} \| \Omega * \left(X - \left(A_{(1)}, A_{(2)}, \ldots, A_{(n)} \right) \right) \|_F^2 \ .$$

3.2 Establish Tensor Model

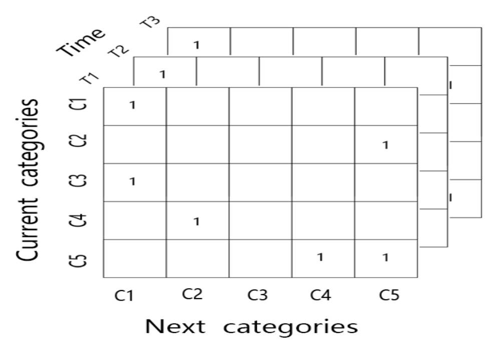
This paper will introduce the four orders of current location category, next location category, time, and season, to model for $x \in R^{T \times C_1 \times C_2 \times S}$,using first-order Markov to predict the next location category, saving time cost compared to traditional methods, from location to category, we assume that the probability of checking in the category is the probability of each category from the current location. average value. Then, the first-order transition probability between the current category and the next category can be formalized as

$$P\left(c_{(lt+1)} \middle| C_{(lt)}\right) = \frac{1}{\left|C_{(lt)}\right|} \sum_{c_{(lt)} \in C_{(lt)}} P(c_{lt+1} | c_{lt}) \tag{3}$$

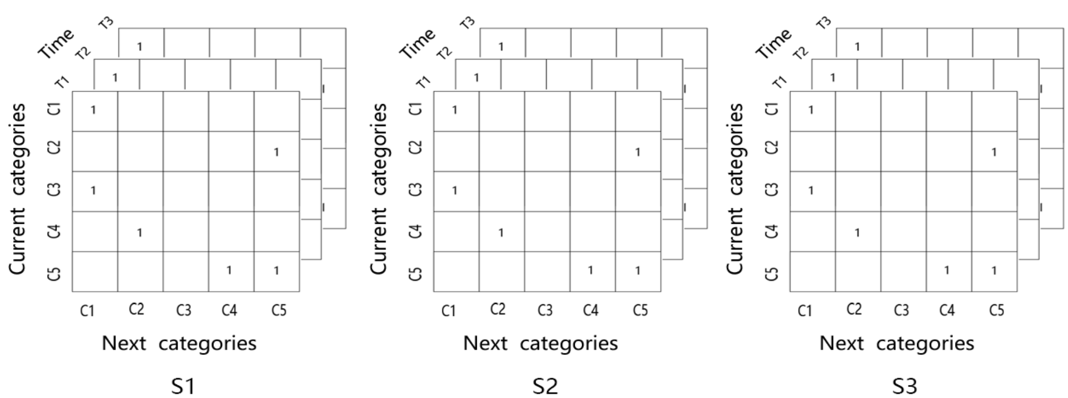
In order to better manage the time series, the time is divided into six intervals T = {T1, T2, T3,T4,T5,T6}, T1 = {2,3,4}, T2 = {5,6,7,8,9}, T3 = {10,11,12,13,14}, T4 = {15,16,17,18,19}, T5 = {20,21,22}, T6 = {23,24,1}, in the modeling, the existing data has been set to “1”, no is blank, we changed the probability equation into the training model.



(a) 2-order tensor--current category and next category



(b) 3-order tensor---current category, next categories and time



(c) 4-order tensor----Current category and next category and time, season

Figure 1: Fourth-order tensor model

For the problem of tensor loss, we add a regularizer to the traditional tensor complement method to improve the accuracy of the prediction, and the objective function becomes

$$\min_{X} f(X)+\lambda r(X) \tag{4}$$

Here r is a low-rank regularizer, λ is a hyperparameter. By combining the non-convex regularizer and the overlapped nuclear norm, we turn Eq. (4) into

$$\min_{x} F(x) \equiv \frac{1}{2}\|P_{\Omega}(x-O)\|_F^2 + \sum_{d=1}^{D}\frac{\lambda_d}{D}\phi(x_{\langle d\rangle}) \tag{5}$$

For a tensor X of order M, the overlapped nuclear norm is $\|x\|_{\text{overlap}} = \sum_{m=1}^{M}\lambda_m\|x_{<m>}\|_*$,

$$\phi(x)=\sum_{i=1}^{n}\kappa(\sigma_i(x)) \tag{6}$$

3.3 Solve Using a Proximal Algorithm

We use the proximal average algorithm to find the optimal solution of Eq. (5). When solving, the problem can be used $Z_t = X_t - \frac{1}{\tau}\nabla f(X_t), \tau > \rho$, τ is the step-size, ρ must be satisfied $\|\nabla f(x)-\nabla f(y)-\|_F^2 \le \rho\|x - y\|_F^2$, $x, y \in R$, $\text{prox}_{\frac{\lambda}{\tau}\|\cdot\|_*}(Z) \equiv \text{Arg}\min_{X}\frac{1}{2}\|X - Z\|_F^2 + \frac{\lambda}{\tau}\|X\|_*$, use the proximal average for Eq. (5).

$$X_t = \frac{1}{D}\sum_{i=1}^{D} y_t^i \tag{7}$$

$$Z_t = X_t - \frac{1}{\tau}P_{\Omega}(X_t - O) \tag{8}$$

$$y_{t+1}^i = \left[\text{prox}_{\frac{\lambda_i}{\tau}}([Z_t]_{<i>})\right]^{<i>} \tag{9}$$

In Eq. (8), multiple tensor fold and unfold are required. To deal with this problem, we maintain a low-rank plus sparse structure. Let Eq. (9) always keep low-rank, where $Z_t^i = [Z_t]_{\langle i\rangle}$, let its rank be κ_t^i. In each iteration, let $Y_t^i = U_t^i(V_t^i)^T$, where $U_t^i \in R^{I_i\times\kappa_t^i}$ and $V_t^i = R^{(I_i \notin I)\times k_t^i}$, after final finishing and induction

$$X_t = \frac{1}{D}\sum_{i=1}^{D}\left[Y_t^i = U_t^i(V_t^i)^T\right]^{<i>} \tag{10}$$

$$Z_t = \frac{1}{D}\sum_{i=1}^{D}\left[Y_t^i = U_t^i(V_t^i)^T\right]^{<i>} - \frac{1}{\tau}P_{\Omega}(X_t - O) \tag{11}$$

3.4 Use of Momentum

In Section 3.3, the specific execution process and iteration formula have been derived from the proximal average algorithm, convergence rate that can be achieved $O(\frac{1}{\xi})$,where $\xi = f(x^{(k)}) - f(x^*)$, i.e., the error between the current iteration result and the optimal solution, by accelerating the algorithm, you can convergence rate of $O(\frac{1}{\sqrt{\xi}})$.

In the process of solving Eq. (5), considering all kinds of optimizers in machine learning, the adaptive momentum ADAM, momentum optimization method is changed based on the gradient drop method, which can accelerate the gradient decline. In this paper, adaptive momentum is used to update the learning rate to improve the training speed. The specific ADAM algorithm is as follows:

Algorithm 1: ADAM

Init $m_0 = 0$ $v_0 = 0$ $t = 0$;
while θ_t not converged do
$t = t + 1$;
$g_t = \nabla_\theta f_t(\theta_{t-1})$;
$m_t = \beta_1 \times m_{t-1} + (1-\beta_1)$;
$v_t = \beta_2 \times v_{t-1} + (1-\beta_2) g_t^2$;
$\hat{m}_t = m_t / (1-\beta_1^t)$;
$\hat{v}_t = v_t / (1-\beta_2^t)$;
$\theta_t = \theta_{t-1} - \alpha \times \hat{m}_t / (\sqrt{\hat{v}_t} + \xi)$;
output θ_t

Algorithm 2: Personalized recommendation for POI via Nonconvex Regurlarized Tensor (PRNORT)

Init $x_0 = x_1 = 0, \tau > \rho + DLand \gamma_{1,} p \in (0,1)$;
for $t = 1 \ldots$ T do
$x_{t+1} = \frac{1}{D} \sum_{i=1}^{D} (U_{t+1}^i (V_{t+1}^i)_T)^{<i>}$;
$\bar{x}_t = x_t + \gamma_t (x_t - x_{t-1})$;
if $F_\tau(\bar{x}_t) <= F_\tau(x_t)$ *then*
$v_t = \bar{x}_t, \gamma_{t+1} = \min(\frac{\gamma_t}{p}, 1)$;
else
$v_t = x_t, \gamma_{t+1} = p\gamma_t$;
end if
$Z_t = v_t - \frac{1}{\tau} P_\Omega(v_t - O)$;
for $i = 1 \ldots D$ do
$X_{t+1}^i = prox_{\frac{\lambda_i}{\tau}}((Z_t)_{<i>})$;
end for
end for
output X_{t+1}

4 Experiment and Analysis

4.1 Data sets and Evaluation Criteria

The data comes from FourSquare, which includes data from January 2010 to June 2011, including New York and Los Angeles, with 249 statistical categories, we will use 80% of the data for training, 20% The data is tested. The data is as follows:

Table 1: Data statistics

City	User	Location	Category	Tip
New York	2,581	206,416	249	166,530
Los Angeles	1,604	215,614	249	109,526

We use the current popular and widely used evaluation standards

$$\text{Precision} = \frac{a}{N} \tag{12}$$

Precision is the ratio of the predicted location category result a to the selection of N location categories.

4.2 Experimental Results and Analysis

In the experimental part, we use PRNORT and MF,PMF,FPMC,CD,TD,TA-FPMC, TAD-FPMC and PRIDO methods to predict the two cities of NY (New York) and LA (Los Angeles), respectively. Due to the addition of time decay factor, it is impossible to calculate the accurate value and the operation is tedious. It can be seen intuitively in Tabs. 3 and 4 that the accuracy of the advanced category prediction method TAD-FPMC is obviously not as good as that of PRNORT.

Table 2: Category prediction method

Model	Scale	Description
MF	$\|U\| \times \|C\|$	a traditional method of using cooperative filtering
PMF	$\|U\| \times \|C\| \times \|C\|$	a personalized recommendation method based on matrix decomposition
FPMC	$\|U\| \times \|C\| \times \|C\|$	the method of adding the first-order Markov chain for the first time
TA-FPMC	$\|U\| \times \|T\| \times \|C\| \times \|C\|$	the first method of modeling using fourth-order tensor
PRIDO	$\|U\| \times \|T\| \times \|C\| \times \|C\|$	modeling using dynamic tensor
TAD-FPMC	$\|U\| \times \|T\| \times \|C\| \times \|C\|$	the fourth-order tensor add time-delay factors
CD	$\|U\| \times \|T\| \times \|C\| \times \|C\|$	a traditional method of tensor decomposition
TD	$\|U\| \times \|T\| \times \|C\| \times \|C\|$	the product of decomposing the tensor into a core and a dimension

Table 3: Results of the category recommendations in the city of New York

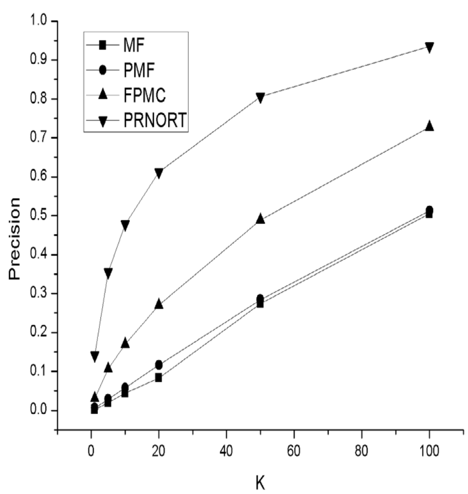
Metrics	MF	PMF	FPMC	CD	TD	TA-FPMC	PRIDO	TAD-FPMC	PRNORT
P (N = 1)	0.0016	0.0060	0.0310	0.0767	0.0921	0.0747	0.1310	0.1230	0.1411
P (N = 5)	0.0197	0.0283	0.1063	0.2221	0.2642	0.2298	0.2454	0.2996	0.3544
P (N = 10)	0.0444	0.0571	0.1700	0.3249	0.3863	0.3397	0.3768	0.4136	0.4783
P (N = 20)	0.0822	0.1160	0.2699	0.4829	0.5357	0.4801	0.5387	0.5615	0.6124
P (N = 50)	0.2744	0.2843	0.4893	0.7193	0.7552	0.7130	0.7488	0.7699	0.8057
P (N = 100)	0.5053	0.5127	0.7280	0.8887	0.9040	0.8812	0.8876	0.8965	0.9355

Table 4: Results of the category recommendations in the city of Los Angeles

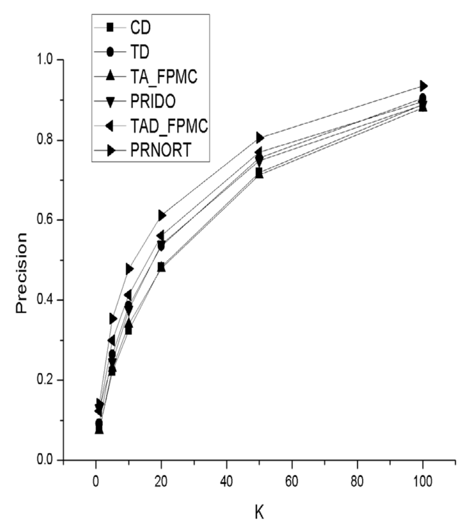
Metrics	MF	PMF	FPMC	CD	TD	TA-FPMC	PRIDO	TAD-FPMC	PRNORT
P (N = 1)	0.0433	0.0057	0.0477	0.0677	0.0964	0.0928	0.1252	0.1519	0.1764
P (N = 5)	0.1142	0.0336	0.1351	0.2270	0.2695	0.2580	0.2741	0.3250	0.3658
P (N = 10)	0.1734	0.0666	0.1992	0.3216	0.3957	0.3610	0.3923	0.4382	0.4709
P (N = 20)	0.2863	0.1305	0.3023	0.4920	0.5477	0.4974	0.5523	0.5756	0.6371
P (N = 50)	0.4486	0.2949	0.5088	0.7242	0.7588	0.7262	0.7598	0.7753	0.8257
P (N = 100)	0.5834	0.5389	0.7412	0.8920	0.9027	0.8839	0.8856	0.8971	0.9142

Tab. 3 shows the accuracy of nine methods performed using the foursqure dataset in New York City. When n = 1,5,10,20,50,100, our PRNORT method is 15%, 19.4%, 15.6%, 9%, 5% and 4% higher than TAD-FPMC respectively. When n is small, the accuracy of category recommendation is low because of the randomness of data. When n = 20, it can be seen from the table that the accuracy of category recommendation is greatly improved. Tab. 4 shows that in Los Angeles City, when n = 1,5,10,20,50,100, PRNORT method is 16%,12.5%,7.5%, 10.6%, 6.5% and 2% higher than TAD-FPMC respectively. When n = 100, the accuracy of CD, TD and TAD-FPMC is about 90%, while PRNORT method is 1% more accurate than the highest TD method.

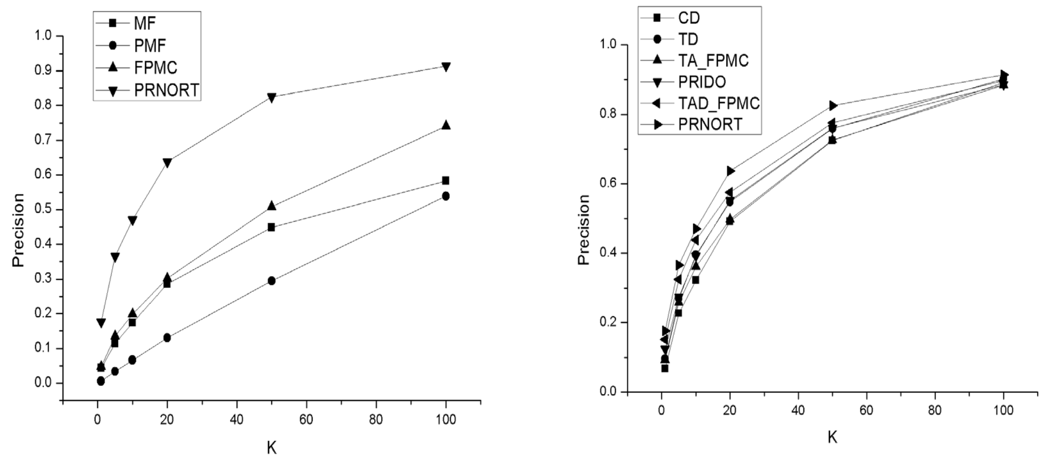
In Fig. 2, (a) indicates the contrast between the PRNORT method and the matrix decomposition method in NY city. (b) is a comparison with (a) the PRNORT method and the tensor decomposition method in the same city. (c) is a comparison of the PRNORT method with the MF method in LA cities. (d) is the case in LA cities where the PRNORT method is compared to the TF method. from (a), (b), (c), (d), we can clearly see that the method of tensor is more accurate than the method of matrix. Our method adds the regularizer to the objective function of the tensor complement to make the result closer to the true Real value. Our seasonal dimension attributes can better speculate the long-term and short-term preferences of users than the user attribute dimensions in the TAD-FPMC method.



(a)



(b)



(c) (d)

Figure 2: Performance comparison of MF, TF and PRNORT in NY and LA cities

5 Conclusion

In this paper, a low-rank tensor completion algorithm based on non-convex regularizer is proposed to improve the accuracy of POI recommendation, and the proximal algorithm is used to find the optimal solution in the objective function. In the whole iterative process, a special sparse plus low-rank structure is added to keep the prediction accuracy. To accelerate the empirical convergence, an adaptive momentum method is added. PRNORT models by constructing 4-dimensional tensor. The 4-dimensions are current location category, next predicted location category, time and season. It forecasts the possible categories of the next location, and then uses the location ranking formula to get the possible location ranking of interest points. The ranking algorithm considers the constraints of popularity and distance. The experimental results based on the Foursquare data set show that our algorithm can guarantee high accuracy while maintaining high efficiency compared with the most advanced models, and the effectiveness of the scheme is proved by a large number of experiments.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

[1] C. Cheng, H. Yang, I. King and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *Proc. of Twenty-Sixth AAAI Conf. on Artificial Intelligence*, Toronto, Ontario, Canada, 2012.

[2] J. D. Zhang, C. Y. Chow and J. Xu, "Enabling kernel-based attribute-aware matrix factorization for rating prediction," *IEEE Transactions on Knowledge Data Engineering*, vol. 29, no. 4, pp. 798–812, 2017.

[3] A. Cichocki, R. Zdunek, A. H. Phan and S. Amari, *Non-Negative Matrix and Tensor Factorizations-Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*, Wiley, Hoboken, NJ, USA, 2009.

[4] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[5] X. Li, M. Jiang, H. Hong and L. Liao, "A time-aware personalized point-of-interest recommendation via high-order tensor factorization," *ACM Transactions on Information Systems*, vol. 35, no. 4, pp. 1–23, 2017

[6] S. Rendle, C. Freudenthaler and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proc. of 19th Int. Conf. on World Wide Web, WWW 2010*, pp. 811-820, Raleigh, NC,

USA, 2010.

[7] S. Zhao, T. Zhao, H. Yang, M. R. Lyu and I. King, “STELLAR: Spatial-temporal latent ranking for successive point-of- interest recommendation,” in *Proc. of Rirtieth AAAI Conf. on Artificial Intelligence*, pp. 315–322, Phoenix, AZ, USA, 2016.

[8] Y. Koren and R. Bell, “Advances in collaborative filtering,” *Recommender Systems Handbook*, pp. 145–186, 2015.

[9] Y. Zheng and Z. Zhang, “Mining interesting locations and travel sequences from GPS trajectories,” in *Proc. of the 18th Int. Conf. on World Wide Web*, ACM, pp. 791–800, 2009.

[10] H. Ye and Z. Zhu, “Whats your next move: User activity prediction in location-based social networks.” in *Proc. of SIAM Int. Conf. on Data Mining,* 2013.

[11] M. Ye and J. Park, “Location-based recommendation system using bayesian user’s preference model in mobile devices,” in *Proc. Ubiquitous Intelligence and Computing, 4th Int. Conf.*, pp. 1130–1139, Hong Kong, China, 2007.

[12] M. Ye, X. Liu and W. C. Lee. “Exploring social influence for recommendation: a generative model approach,” *SIGIR*, pp. 671–680, 2012.

[13] E. Cho, S. A. Myers and J. Leskovec, “Friendship and mobility: User movement in location-based social networks,” *KDD*, pp. 1082–1090, 2011.

[14] Y. Liu and W. Wei, “Exploiting geographical neighborhood characteristics for location recommendation.” in *Proc. of ACM Int. Conf. on Information and Knowledge Management*, pp.739–748, 2014.

[15] D. Lian and C. Zhao, “Joint geographical modeling and matrix factorization for point-of-interest recommendation,” in *Proc. of ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 831–840, 2014.

[16] W. Q. Wang, H. Yin and C. Ling, “ST-SAGE: A spatial temporal sparse additive generative model for spatial item recommendation,” *ACM TIST 8*, vol. 48, pp. 1–25, 2017.

[17] H. Wang and H. Shen, “Exploiting POI-specific geographical influence for point-of-interest recommendation,” in *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence*, pp. 3877–3883,2018.

[18] M. Ye, “What you are is when you are: The temporal dimension of feature types in location-based social networks.” in *Proc. of ACM Sigspatial Int. Conf. on Advances in Geographic Information Systems*, pp. 102–111, 2012.

[19] B. Chang, “Content-aware hierarchical point-of-interest embedding model for successive POI recommendation,” in *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence*, pp. 3301–3307, 2018.

[20] H. Yin, “Joint modeling of user check-in behaviors for real-time point-of-interest recommendation,” *ACM Transactions on Information Systems*, vol. 35, no. 2, pp. 1–44, 2016.

[21] H. Yin, “A location-content-aware recommend system.” in *Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 221–229, 2013.

[22] H. Yin, “Spatial-aware hierarchical collaborative deep learning for POI recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, vol 29. pp. 2537–2551, 2017.

[23] J. Liao, J. Tang and X. Zhao, “Improving POI recommendation via dynamic tensor completion,” *Scientific Programming*, pp. 1–11, 2018.