AUC-MF: Point of Interest Recommendation with AUC Maximization

AUC-MF: Point of Interest Recommendation with AUC Maximization

Peng Han[1,2], Shuo Shang[3], Aixin Sun[2], Peilin Zhao [4], Kai Zheng[5], Panos Kalnis[1]
[1] *King Abdullah University of Science and Technology, Saudi Arabia*
{peng.han, panos.kalnis}@kaust.edu.sa
[2] *School of Computer Science and Engineering, Nanyang Technological University, Singapore*
AXSun@ntu.edu.sg
[3] *Inception Institute of Artificial Intelligence*
jedi.shang@gmail.com
[4] *Tencent AI Lab, China*
masonzhao@tencent.com
[5] *Big Data Research Center, University of Electronic Science and Technology of China, China*
zhengkai@uestc.edu.cn

***Abstract*—The task of point of interest (POI) recommendation aims to recommend unvisited places to users based on their check-in history. A major challenge in POI recommendation is data sparsity, because a user typically visits only a very small number of POIs among all available POIs. In this paper, we propose AUC-MF to address the POI recommendation problem by maximizing Area Under the ROC curve (AUC). AUC has been widely used for measuring classification performance with imbalanced data distributions. To optimize AUC, we transform the recommendation task to a classification problem, where the visited locations are positive examples and the unvisited are negative ones. We define a new lambda for AUC to utilize the LambdaMF model, which combines the lambda-based method and matrix factorization model in collaborative filtering. Experiments on two datasets show that the proposed AUC-MF outperforms state-of-the-art methods significantly in terms of recommendation accuracy.**

I. Introduction

Location-based social networks (LSBNs) allow users to check in and share their experiences when they visit a point of interest (POI), such as a museum or a restaurant. With the development and popularity of various LSBN (Fig. 1) platforms *e.g.,* BrightKite, Foursquare, and Gowalla, user check-in data is growing at an unprecedented pace. For instance, Foursquare had more than 50 million active users and more than 8 billion check-ins made by 2016. The availability of abundant amount of user check-in data, enables many studies on recommender systems to further enhance user experiences. POI recommendation aims at finding unvisited locations that a user may be interested in, by learning from users check-in history and other related factors. POI recommendation is challenging for many reasons. One of the most important reasons is that user check-in data is *extremely sparse*.

In this paper, we propose to maximize AUC to handle data sparsity in POI recommendation. AUC is a metric that is widely used for measuring classification accuracy for imbalanced data distributions. To the best of our knowledge, we are the first to use AUC as the objective function for POI recommendation. In this work, we utilize the framework of LambdaMF [1], which combines lambda-based method [2] and the highly-successful matrix factorization model in collaborative filtering. More specifically, we use AUC as the objective function and define a new lambda which could utilize the property of AUC in our new framework named AUC-MF. Experimental results show that AUC-MF achieves state-of-the-art accuracy.

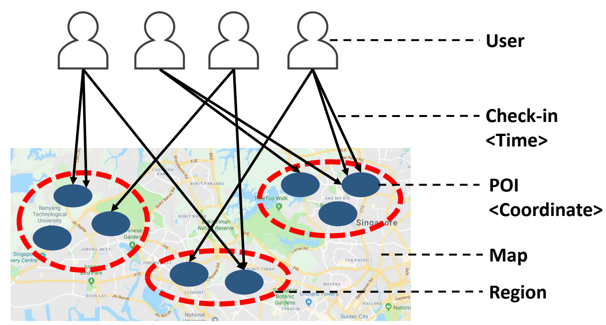

Fig. 1. Location-based social network

We summarize our contribution in this paper as follows:

- We propose a new framework named AUC-MF for POI recommendation, which effectively addresses the data scarcity problem.
- We define a new lambda for our AUC-MF to get the gradients.
- Extensive experiments on the dataset from Gowalla demonstrate that AUC-MF outperforms state-of-the-art methods significantly for POI recommendation.

The rest of the paper is organized as follows. Section II introduces related works. Section III gives the preliminary knowledge about POI recommendation. Section IV presents our recommendation framework. After reporting experiments in Section V, we conclude this paper in Section VI.

II. Related Work

A. Matrix Factorization

Matrix Factorization (MF) [3], the most popular algorithm in the field of recommendation, decomposes check-in matrix into user matrix and POI matrix. IRenMF [4] is based on Weighted Matrix Factorization (WMF) [5], [6]. Rank-GeoFM [7] proposes a pair-wise non-smooth loss to reduce the influence of data sparsity problem. LRT [8] is a time-based matrix factorization model.

B. Non-smooth Function

Non-smooth quality measures are particularly difficult to optimize directly, because the model scores are computed with the ranks of items. In [2], they propose a general framework to optimize non-smooth function with stochastic gradient descent method. LambdaMF [1] combines the Lambda method [2] with matrix factorization to solve the movie recommendation problem. AdaBPR [9] is a boosting framework, which is focused on implicit recommendation.

III. Preliminaries

A. Problem Definition

In this section, we firstly give concepts and their notations, after which we define the problem of POI recommendation.

Given the historical interactions between m users and n POIs, the task of POI recommendation is to recommend a target user u a list of POIs. The POIs in the recommended list should have not been visited by the target user. In many real-world scenarios, POI recommendation tasks are based on user implicit preference feedback, *i.e.,* whether a user has visited a POI. This kind of feedback is usually represented by using a set of binary variables $y_{ui} \in \{0, 1\}$. If a user u has visited a POI i, y_{ui} is set to 1, and 0 otherwise. Note that $y_{ui} = 0$ does not explicitly indicate that u is not interested in i. It may be the result that user u does not know the existence of i. In this paper, we denote the set of users and POIs by U and L, respectively. For a user u, we denote her visited POIs by $L_u^+ = \{i | y_{ui} = 1, i \in L\}$ and denote her unvisited POIs as $L_u^- = L \setminus L_u^+$. Then, the set of all user-POI interactions is defined as $\mathfrak{D} = \{(u, i) | u \in U, i \in L_u^+\}$. For POI recommendation, we aim to recommend every user u a list of POIs from their unvisited POIs L_u^-.

B. Overview of the Proposed Solution

To address the data sparsity of POI recommendation, we reformulate recommendation into an imbalanced classification problem. AUC optimization is then applied to solve this problem. To handle the non-smooth component in the AUC formula, Lambda-based method is used to optimize the objective function. To make the optimization be consistent with AUC, we define a new lambda with AUC definition. Combining with matrix factorization, our method for POI recommendation is named AUC-MF.

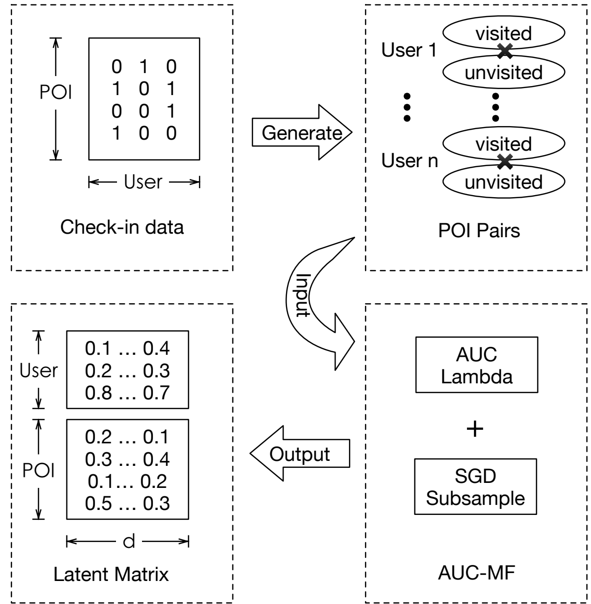

Fig. 2. Overview of AUC-MF. Firstly we generate positive and negative user-POI pairs from the check-in data. Then we use them as the input to AUC-MF. By optimizing AUC-MF with stochastic gradient descent (SGD) method, we get the user preference matrix.

IV. AUC-MF for POI Recommendation

This section details the proposed AUC-MF, illustrated in Fig. 2. Given the check-in data, we generate positive and negative user-POI pairs, that would be used as the input to AUC-MF. By optimizing AUC-MF with stochastic gradient descent (SGD) method, we get the user preference matrix.

A. AUC Metric for POI Recommendation

AUC [10] is a decision threshold metric that the probability for a randomly drawn positive instance has a higher decision value than a randomly sampled negative instance. With the definition of AUC measure for binary classification [11], we define the AUC measure for POI recommendation as follows:

$$\begin{aligned}\mathrm{AUC} &= \frac{1}{|U|}\sum_{u\in U}\frac{1}{|L_u^+|}\sum_{i\in L_u^+}\frac{1}{|L_u^-|}\sum_{j\in L_u^-}\mathbb{I}(\pi_{ui} > \pi_{uj}) \\ &= 1 - \frac{1}{|U|}\sum_{u\in U}\frac{1}{|L_u^+|}\sum_{i\in L_u^+}\frac{1}{|L_u^-|}\sum_{j\in L_u^-}\mathbb{I}(\pi_{ui} \le \pi_{uj})\end{aligned} \tag{1}$$

In the above equation, $|\cdot|$ denotes the cardinality of the set. π_{ui} is the rank position of i in the ranked POI list, generated based on the user u's preferences. The rank is based on the predicted scores $(s_1^u, s_2^u, ..., s_n^u)$, in descending order. $\mathbb{I}(\cdot)$ is an indicator function that outputs 1 if the condition holds and 0 otherwise. Thus maximizing AUC is equivalent to minimizing

$$C = \sum_{u\in U}\frac{\sum_{i\in L_u^+}\sum_{j\in L_u^-}\mathbb{I}(\pi_{ui} \le \pi_{uj})}{|L_u^+||L_u^-|}. \tag{2}$$

For any $(u, i) \in \mathfrak{D}$ and $j \in L_u^-$, we define the cost function $C_{i,j}^u$ as follows:

$$C_{i,j}^u = \frac{\mathbb{I}(\pi_{ui} \leq \pi_{uj})}{|L_u^+||L_u^-|} \tag{3}$$

Thus,

$$C = \sum_{u \in U} \sum_{i \in L_u^+} \sum_{j \in L_u^-} C_{i,j}^u \tag{4}$$

However, the indicator function $\mathbb{I}(\cdot)$ is a non-smooth function, which means that the derivatives of the cost with respect to the model parameters are either zero or undefined. To apply SGD for optimization, we need to generate gradients for the cost function. Next, we will focus on the virtual derivatives of the cost $C_{i,j}^u$ with respect to the model parameter w.

B. Rank with Non-smooth Function

LambdaRank [2] provides a method that can be extended to any non-smooth and multivariate cost functions. Taking the POI recommendation as an example, for user u, the gradient of an implicit cost function $\widetilde{C}^u$ with respect to the score s_j^u of POI j is written as

$$\frac{\partial \widetilde{C}^u}{\partial s_j^u} = -\lambda_j^u(s_1^u, y_{u1}, ..., s_n^u, y_{un}). \tag{5}$$

We give the general form of the lambda method for POI recommendation with $(u, i) \in \mathfrak{D}$ and $j \in L_u^-$.

$$\frac{\partial C_{i,j}^u}{\partial s_i^u} = -\lambda_{i,j}^u = -\frac{\partial C_{i,j}^u}{\partial s_j^u} \tag{6}$$

When $\lambda_{i,j}^u$ is set as a positive constant, POI i must move up and POI j must move down in the ranked list to reduce the cost. Then the derivatives of the cost $C_{i,j}^u$ with respect to the model parameter w is

$$\begin{aligned} \frac{\partial C_{i,j}^u}{\partial w} &= \frac{\partial C_{i,j}^u}{\partial s_i^u}\frac{\partial s_i^u}{\partial w} + \frac{\partial C_{i,j}^u}{\partial s_j^u}\frac{\partial s_j^u}{\partial w} \\ &= -\lambda_{i,j}^u \frac{\partial s_i^u}{\partial w} + \lambda_{i,j}^u \frac{\partial s_j^u}{\partial w} \end{aligned} \tag{7}$$

After we get the derivatives, we can apply them in the matrix factorization based methods to tackle the task of POI recommendation.

C. AUC Lambda

In the previous sections we have provided a general definition of λ, and we now discuss how to choose λ explicitly. There are two conditions λ should satisfy, that would make the implicit cost function exist and be convex. First, the Jacobian matrix of the implicit cost function with respect to the rating score must be symmetric. This represents that there exists a cost function for which λ is derivative. Once satisfying the condition of existence, we should make sure that the implicit cost function is convex. That is, the Jacobian matrix should be positive semidefinite everywhere. As discussed before, the constant λ could satisfy both conditions. Considering minimizing the cost function, λ should be positive. Therefore, given any $(u, i) \in \mathfrak{D}$ and $j \in L_u^-$, we set

$$\lambda_{i,j}^u = |\Delta\text{AUC}| = \frac{|r_i^u - r_j^u| + 1}{|L_u^+||L_u^-|} \tag{8}$$

where r_i^u denotes the predicted rank of POI i for user u and ΔAUC means the absolute AUC difference by swapping the two POIs. The whole process of AUC-MF is given in Algorithm 1.

Algorithm 1 AUC-MF

Input: The observed interactions D, learning rate η, regularization weight α, and number of iterations n_iter
Output: The learned user latent matrix P and POI latent matrix Q

Initialize P and Q randomly;
for $t = 1$ to n_iter **do**
  for $u = 1$ to m **do**
    $\frac{\partial \hat{C}^u}{\partial p_u} \leftarrow 0$;
    for $i = 1$ to n **do**
      $\frac{\partial \hat{C}^u}{\partial q_i} \leftarrow 0$;
      $r = argsort(p_u * Q^\top)$
      for all $(u, i) \in \mathfrak{D}$ **do**
        Randomly draw an POI j from L_u^-;
        $\lambda = \frac{|r_i - r_j| + 1}{|L_u^+|}$
        $\frac{\partial \hat{C}^u}{\partial p_u} += \lambda(q_j - q_i) + \alpha p_u$
        $\frac{\partial \hat{C}^u}{\partial q_i} += -\lambda p_u + \alpha q_i$
        $\frac{\partial \hat{C}^u}{\partial q_j} += \lambda p_u + \alpha q_j$
      end for
      $p_u -= \eta \frac{\partial \hat{C}^u}{\partial p_u}$
      for $i = 1$ to n **do**
        $q_i -= \eta \frac{\partial \hat{C}^u}{\partial q_i}$
      end for
    end for
  end for
end for
return P, Q

D. Subsample

For movie recommendation, unlabeled data is not needed in the training process. Different from that, POI recommendation with AUC metric would put all the POI pairs in the optimization process. Doing so would make the optimization of our algorithm extremely time-consuming. To handle this problem, we randomly draw a POI j from L_u^- with uniform probability $\frac{1}{|L_u^-|}$ for every $(u, i) \in \mathfrak{D}$. That is also the reason that the component $\frac{1}{|L_u^-|}$ does not appear in the updating of model parameter λ.

V. Experiments

In this section, we empirically evaluate AUC-MF against state-of-the-art methods that use the same settings as ours.

TABLE I
RESULTS ON NEW YORK DATASET

Method	Pr	Re
BRPMF [15]	4.17%±0.06%	5.47%±0.07%
AoBRP [16]	4.33%±0.02%	6.29%±0.06%
WRMF [4]	5.63%±0.07%	7.00%±0.11%
LRT [8]	5.62%±0.08%	6.49%±0.13%
IRenMF [4]	5.75%±0.04%	7.10%±0.03%
RankGeoFM [7]	5.29%±0.12%	6.17%±0.16%
AUC-MF	**5.86%**±0.02%	**7.82%**±0.03%

A. Experimental Settings

1) Dataset: We use the dataset that has been used in [4], where the authors collected check-in data from Gowalla from November 2010 to June 2011. There are in total 36,001,959 check-ins made by 319,063 users over 2,844,076 locations. Each check-in contains user id, location id and timestamp. Latitude and longitude are available for all locations. To estimate the performance of our framework, we construct two datasets by extracting check-in data for New York city with reverse-geocoder[1].

In our experiments, the dataset is divided into three parts. For each user, we sort the check-ins by timestamp, and choose the first 70% as training set, then 10% as development set, and the remaining 20% as test set.

2) Evaluation Metrics: We evaluate the quality of POI recommendation by two metrics, Precision (Pr) and Recall (Re), as in most other works [12], [13]. Given a user u, L_u^T denotes the set of corresponding visited locations in the testing data, and L_u^R denotes the set of recommended locations by the algorithm. The definitions of precision and recall are:

$$\begin{aligned} Pr &= \frac{1}{|U^T|} \sum_{u \in U^T} \frac{|L_u^T \cap L_u^R|}{K} \\ Re &= \frac{1}{|U^T|} \sum_{u \in U^T} \frac{|L_u^T \cap L_u^R|}{|L_u^T|} \end{aligned} \tag{9}$$

where U^T is the set of users in the testing dataset and $K = 10$ in our experiments.

3) Evaluated Methods: We compare the performances of many state-of-the-art methods in Table. I. Two of them utilize geographical context and achieve the top two performance in the latest experimental evaluation of POI recommendation task [14] (see Section II for more details).

B. Experiments Results

The recommendation accuracies of AUC-MF and other baseline methods are summarized in Table I. From the results, we could see that AUC-MF outperforms all state-of-the-art baselines on this dataset on both metrics precision and recall.

VI. CONCLUSION

In this paper, we propose a new framework, named AUC-MF, for POI recommendation with AUC maximization. To optimize the AUC metric, we utilize lambda-based method to generate an implicit cost function for this non-smooth function. To satisfy the two conditions of that method, we define a new constant lambda for AUC, which could make the implicit cost function exist and ensure it is convex. Then we combine the lambda-based method with matrix factorization as LambdaMF. We use SGD to optimize the cost function and draw random negative samples with universal distribution to reduce the optimizing time. Experiments on real data from New York show that our AUC-MF outperforms state-of-the-art baselines.

[1] https://github.com/thampiman/reverse-geocoder

REFERENCES

[1] G.-H. Lee and S.-D. Lin, "Lambdamf: Learning nonsmooth ranking functions in matrix factorization using lambda," *ICDM*, 2015.

[2] C. J. C. Burges, R. Ragno, and Q. V. Le, "Learning to rank with nonsmooth cost functions," in *NIPS*, 2006.

[3] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, 2009.

[4] Y. Liu, W. Wei, A. Sun, and C. Miao, "Exploiting geographical neighborhood characteristics for location recommendation," in *CIKM*, 2014.

[5] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," 2008.

[6] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. M. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," 2008.

[7] X. Li, G. Cong, X. Li, T.-A. N. Pham, and S. Krishnaswamy, "Rank-geofm: A ranking based geographical factorization method for point of interest recommendation," in *SIGIR*, 2015.

[8] H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring temporal effects for location recommendation on location-based social networks," in *RecSys*, 2013.

[9] Y. Liu, P. Zhao, A. Sun, and C. Miao, "A boosting algorithm for item recommendation with implicit feedback," in *IJCAI*, 2015.

[10] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve." *Radiology*, 1982.

[11] Y. Ying, L. Wen, and S. Lyu, "Stochastic online auc maximization," in *NIPS*, 2016.

[12] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *AAAI*, 2012.

[13] M. Ye, P. Yin, W.-C. Lee, and D. L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *SIGIR*, 2011.

[14] Y. Liu, T.-A. Pham, G. Cong, and Q. Yuan, "An experimental evaluation of point-of-interest recommendation in location-based social networks," *PVLDB*, 2017.

[15] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*, 2009.

[16] S. Rendle and C. Freudenthaler, "Improving pairwise learning for item recommendation from implicit feedback," in *WSDM*, 2014.