Modeling Personalized Out-of-Town Distances in Location Recommendation

Daizong Ding*, Mi Zhang*‡, Xudong Pan*, Min Yang*, Xiangnan He†
*School of Computer Science, Fudan University
†School of Data Science, University of Science and Technology of China
Emails: {17110240010, mi_zhang, 18110240010, m_yang}@fudan.edu.cn, xiangnanhe@gmail.com

***Abstract*—Location recommendation becomes increasingly important in the mobile era. Particularly, how to exploit personalized geographical preferences determines the quality of recommended results. A number of efforts have been made on this task, however, there exists a common limitation called the out-of-town recommending problem, i.e., those far places can hardly be recommended. In this paper, we first reveal why modeling the geographical patterns is difficult with the help of the extreme value theory. We find that out-of-town distances are heavy-tailed variables with few observations and extreme values, making it difficult to use common distributions to describe them. To address this issue, we propose a new function called volcano function to model out-of-town distances and personalize it for different users. Empirical results show that we can learn effective patterns from limited observations. Finally we extend the volcano function to a ranking-based collaborative filtering framework, naming it as volcano network (VolNet). Experimental results show the superior performance of VolNet, especially the recall is improved from 0.2 to 0.35 in recommending remote venues compared with the state-of-the-art method GeoMF++.**

***Index Terms*—Location recommendation, Out-of-Town, Extreme Value Theory**

I. Introduction

As the number of GPS built-in devices grows rapidly, geographical data are widely used and produced in online services, e.g. the check-in data in Meetup. In these services, location recommendation plays a vital role because it can encourage users to explore more places. Nevertheless, it is difficult to find Point-of-Interests (POIs) among numerous venues due to the sparsity of user-venue matrix [1], [2]. For instance, there are more than 105 million venues in Foursquare until 2019, where users only visited 200 of them in average. Towards this problem, several methods have proposed to utilize side information, e.g., geographical coordinates [3], social connections [4] and textual contents [5]. Particularly, approaches developed on geographical coordinates are attracting more and more attention because they can be easily applied in various location recommendation scenarios [6], [7]. Therefore we focus on utilizing geographical information to improve the quality of location recommendation.

According to previous research, the cornerstone of this task is how to better model personalized geographical patterns of visited places [9]–[11]. A number of efforts have been made on it, for instance, using power-law distribution to model the distances between visited venues [3] and detecting geographical

‡ Mi Zhang is the corresponding author.

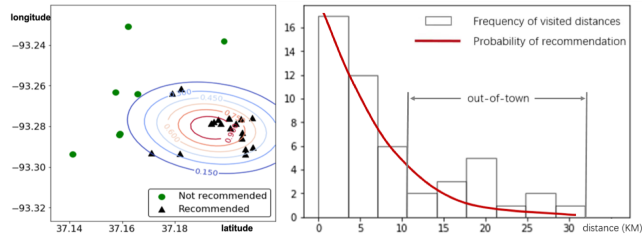


Fig. 1. The left figure is the recommendation results by one state-of-the-art method (GeoMF++ [8]), where contour line depicts the probability of being recommended. The left figure shows the frequency of visited distances of a user. The detailed settings such as how to find the center of the user are listed in Sec. 3.

clusters on visited venues [12]. Although empirical results state the effectiveness of these methods [2], there exists a common concern, that is, they hardly can recommend *out-of-town* places to users [13]. In order to demonstrate this issue, we evaluate the state-of-the-art method GeoMF++ [8] and plot the recommended results for a user in Fig. 1. As we can see from the results, although the user sometimes visit far places, the model fails to recognize this geographical pattern.

These out-of-town venues, which are far from commonly visited places, is important in improving the experience of a location recommender system [14]. For example, a user will be satisfied if being recommended with a good place for hiking at the weekend. According to previous researches, users may visit up to 30% far places in real systems [15]. In order to improve the quality of recommended venues, several works propose to use contextual data such as temporal and social information to recommend out-of-town venues [1], [13], [15], however, they cannot be generalized to various location recommendation applications because some datasets do not have that kind of side information. Until now, there still lacks the work of modeling geographical patterns of out-of-town venues for better location recommendation. Therefore in this paper, we aim to answer the following questions: **(1)** Why current methods lack the ability of modeling geographical patterns of out-of-town venues? **(2)** How to present a general solution for this problem?

For the first problem, we propose to use extreme value theory (EVT) to reveal the difficulties. In summary, the challenges are mainly two-folds. **(1)** According to EVT, data in real-world applications often follow *heavy-tailed* distributions. However, most distributions that we used are *light-tailed*, which leads to the fact that we often underestimate the probability of

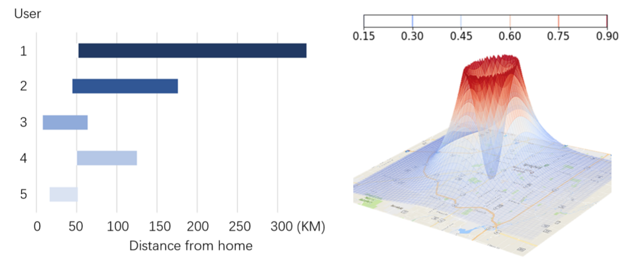


Fig. 2. The left figure demonstrates the range of visited distances from different users. The right figure is the proposed volcano function on the map.

tailed data. For instance, in our problem, the probability of recommending large distances will be much lower than the actual visiting frequency, as shown in Fig. 1; **(2)** Compared with visited distances around the center, the number of tailed distances is much smaller, therefore it is hard to model the tailed distribution with few observations. Considering the difficulties caused by heavy-tailed data and limited observations, how to model the geographical patterns of out-of-town venues is still an open problem.

In this paper, we propose to address this issue in location recommendation. First we assume that, these out-of-town venues should have the geographical property of *neither close nor too far*. The reason of not too far is that people can hardly visit those places due to the intolerable traffic cost [16]. Based on the motivation, we propose a heavy-tailed function called *volcano function* to measure the out-of-town probability of a certain location, whose shape is like a volcano on the map in Fig. 2. Second, we find that users often have personalized out-of-town ranges. As a demonstration, we plot the largest 10% visited distances of five users in Fig. 2. Places around 100km may be out-of-town for user 1, however they are too far to reach for user 5. Therefore we propose to assign each user with a personalized volcano, where the size and steepness vary from person to person. Furthermore, we take the clustering phenomenon into consideration, that is, the visited locations of users often form several areas [17], [18]. For instance, a user may have visited several cities. Based on this phenomenon, we propose to learn multiple personalized volcanoes rather than a single volcano for each user. With the help of EVT, we successfully tackle the above problems in modeling out-of-town distances (heavy-tailed data and limited observations) and learn effective volcanoes for each user. Finally, as the comprehensive framework, we propose a novel location recommendation solution called *Volcano Network* (VolNet), which can take out-of-town distances into consideration during the prediction. In summary, our main contributions are:

- We study out-of-town recommending problem with the EVT explaining why current location recommendation methods suffer from this problem. Furthermore, we propose a new function to model the geographical patterns of out-of-town venues. With the help of EVT, we successfully tackle the problem of heavy-tailed data and limited observations. To our best knowledge, it is the first work to model out-of-town distances in location recommendation.
- We propose VolNet by integrating the proposed function into POI recommendation. We evaluate the effectiveness of VolNet with extensive experiments on three real-world location recommendation datasets (i.e. Foursquare, Yelp and Meetup). Empirical results state that our proposed framework achieves significant improvements compared with the state-of-the-art methods. For instance, we make 80% relative improvement on recall.

II. Related Work

In this section we briefly review the related works. Location recommendation is a crucial task in modern recommendation system. A number of studies exploit geographical preferences to improve the recommendation, e.g. [3] assumes that distances between visited POIs follow the power-law distribution (PD), and recommends those locations with high probability under the estimated distribution. [19] extends the model to the cases with additional temporal information. [18] assumes each user has several clusters on visited POIs and proposed to use Multi-center Gaussian Model (MGM) to model these clusters. [20] extends the PD to Kernel Density Estimator (KDE) with Gaussian Kernel, where [21] extends the KDE to 2-dimensional KDE. [2] further improves the performance of KDE [2], [6], [8] by introducing collaborative filtering technique [22], [23]. There also exist several works using other kinds of side information such as social connections [24], [25], tags [26] and temporal information [27]–[29]. However, they require specific context information and lack the ability of being extended to other location recommendation scenarios [9], [10].

Although current methods achieve promising results, they suffer from the *out-of-town* recommending problem, where the recommended venues are usually near the user [13]. Several works try to recommend out-of-town venues by additional side information. For example, [15] proposes to use social connections to find potential remote venues from similar users. [1] proposes to use text information inside out-of-town venues, for example, park and beach. [14] proposes to find pattern inside visiting timestamps of out-of-town venues. However, as we have discussed above, they strongly rely on the specific side information and they are not able to be applied in other location recommendation applications. Recently, [30] proposes to model out-of-town regions rather than out-of-town venues. However, there still lacks the work of modeling personalized geographical preferences on out-of-town venues due to the sparsity and extreme values.

III. Preliminaries

In this section, we will explain why current methods lack the ability of modeling out-of-town distances under the perspective of extreme value theory.

A. Notations

The set of venues is denoted as $\mathcal{L}$ with size $|\mathcal{L}|$. For each venue $i \in \mathcal{L}$, its geographical coordinate (i.e., longitude and latitude) can be described as $\phi_i \in \mathbb{R}^2$. The set of users is denoted as $\mathcal{U}$ with size $|\mathcal{U}|$. Each user $u \in \mathcal{U}$ contains a list of visited POIs $\mathcal{L}_u$ with size $|\mathcal{L}_u|$. The POI recommendation task

TABLE I
MATHEMATICAL NOTATIONS

Symbol	Size	Description
ϕ_i	$\mathbb{R}^2$	Longitude and latitude for venue i
$\mathcal{L}_u$	$\|\mathcal{L}_u\|$	Venues visited by user u
$\tilde{\mathcal{L}}_{uk}$	$\|\tilde{\mathcal{L}}_{uk}\|$	Venues in cluster k of user u
d_{uik}	$\mathbb{R}$	Distance between venue i and center of user u's cluster k
y_{uik}	$\{0,1\}$	Label of out-of-town distance
K	$\mathbb{R}$	Number of clusters
D	$\mathbb{R}$	Size of latent factors in MF
μ_{uk}	$\mathbb{R}^2$	Center of cluster k of user u
Σ_{uk}	$\mathbb{R}^{2\times 2}$	Covariance of cluster k of user u
ρ_{uik}	$\mathbb{R}$	Probability of location i belongs to cluster k of user u
m_{uk}, λ_{uk}	$\mathbb{R}$	Minima and variance of distances in cluster k of user u ($\tilde{\mathcal{L}}_{uk}$)
α_{uk}, β_{uk}, s_{uk}, b_{uk}	$\mathbb{R}$	Paramters in volcano function for user u's cluster k
γ_{uk}, a_{uk}	$\mathbb{R}$	Parameters for fitting tail distribution
W_u, H_i	$\mathbb{R}^D$	Latent vectors in MF
X_u, Y_u	$\mathbb{R}^K$	Preference vectors on clusters for user u
M_u	$\mathbb{R}^{K\times K}$	Correlation between Peak and Volcano vector for user u.
P_{ui}	$\mathbb{R}^K$	Peak vector of user u and venue i, where P_{uik} is the element of cluster k
V_{ui}	$\mathbb{R}^K$	Volcano vector of user u and venue i, where V_{uik} is the element of cluster k
ξ	$\mathbb{R}_+$	Threshold for filtering far venues from center
ξ_1, ξ_2	$\mathbb{R}_+$	Threshold for generating training samples, where $\xi_1 >> \xi_2$

we focus on is to utilize the observations $\mathcal{L}_u$ to recommend potential POIs for each user u.

B. Calculating Visited Distances

Considering that directly modeling geographical coordinates is complex, current methods commonly transform the coordinates to distances for further study. We also apply such technique in this paper. Given a visited venue set $\mathcal{L}_u$ by user u, we suppose there are K clusters inside visited venues, where this assumption can be easily found in previous works [18]. For each cluster k, we can write its center as $\mu_k \in \mathbb{R}^2$, which represents the geographical coordinate of the cluster. Meanwhile, we can write the covariance of the cluster as $\Sigma_k \in \mathbb{R}^{2\times 2}$, which represents the range of the cluster. The center and covariance vary from person to person. For the sake of simplicity, we omit the subscript u in the rest of this section . For instance, we simplify μ_{uk} to μ_k. Then we can calculate the center and variance by minimizing following objective function:

$$\min_{\rho,\mu,\Sigma} \sum_{i\in\mathcal{L}_u} \sum_{k=1}^{K} \rho_{ik}(\phi_i - \mu_k)^T \Sigma_k^{-1} (\phi_i - \mu_k), \tag{1}$$

where $\rho_{ik} \in [0,1]$ is the probability of location i belonging to cluster k, and $\sum_{k=1}^{K} \rho_{ik} = 1$. Several mature algorithms by iterative optimization can be efficiently applied to solve the optimization problem above [31], and one of the solution can be found in the appendix. By optimizing Eq. 13, the clusters of a user can be characterized by parameter μ and Σ. For instance in Fig. 3(a), with visited venues of a certain user, two corresponding clusters can be learned. In addition, due to the fuzziness of the clustering algorithm brought by parameter ρ, it is reasonable to assume each user has the same cluster size C [17], [18], [32]. Based on the above process, given location $i \in \mathcal{L}_u$, the distance between i and cluster k of can be calculated as $d_{ik} = (\phi_i - \mu_k)^T \Sigma_k^{-1} (\phi_i - \mu_k)$.

C. Why Modeling Out-of-Town Distances is Difficult?

We continue to study the pattern inside distances for each cluster. For instance, why it is difficult to model out-of-town distances. Before that, we should remove venues in the other clusters. For cluster k, we form a new venue set as,

$$\tilde{\mathcal{L}}_{uk} = \{i | i \in \mathcal{L}_u \text{ and } \rho_{ik} > \xi\}, \tag{2}$$

where ξ is a preset small threshold. Different clusters may share the same out-of-town venues, for example in Fig. 3(a), venues with latitudes in $[35.6, 35.65]$ are in both cluster 1 and 2. Then we can study the distribution of visited distances in each cluster. Specifically, we fit them with exponential distribution, which is widely used in modeling visited distances with the following cumulative density function (c.d.f) [3]:

$$1 - F(d) = e^{\frac{d-m_k}{\lambda_k}}, \tag{3}$$

where $m_k \in \mathbb{R}$ and $\lambda_k > 0$ are mean and variance of the distances respectively for cluster k. The inference of them can be found in the appendix. As we can see from the results in Fig. 3(b), although the distribution fits venues around the center well, the probability is lower than the empirical frequency in remote places. Therefore these places are easily be neglected during the prediction.

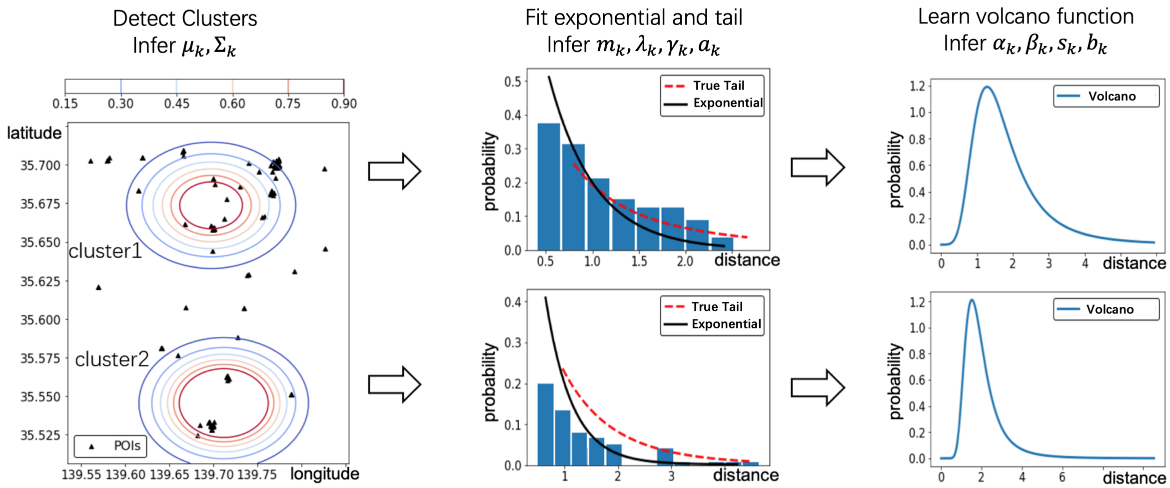
From the view of extreme value theory, this failure comes from the fact that commonly used distributions such as Gaussian, Exponential and Poisson are *light-tailed* distributions. However, data in most real-world applications usually follows *heavy-tailed* distributions [33]. The advantage of light-tailed distributions is that they can model centered data well with the help of the law of large numbers. However, it is hard to apply them on heavy-tailed data mainly because [33]: **(1)** the observations of tailed data is not sufficient enough; **(2)** heavy-tailed data often have unbounded expectation and momentum, which makes it difficult to infer parameters of the distribution. Considering the above challenges, current location recommendation methods use light-tailed distributions to model visited distances, which makes them unable to model out-of-town distances.

IV. MODELING OUT-OF-TOWN DISTANCES

In this section, we will present how to address the issues of limited observations and extreme values in modeling out-of-town distances.

A. Volcano Function

In order to tackle the above issues, we propose to utilize prior knowledge. First, as we have discussed above, those out-of-town distances have the geographical property of *neither close nor too far*. Therefore we can use both geographical property and heavy-tailed property to model out-of-town distances.



(a) Heat map for clustering on a user, where contour line is the probability of belonging to a cluster.
(b) Distribution of distances in each cluster, where the histogram is the empirical distribution.
(c) The shape of volcano function for each cluster.

Fig. 3. Procedures of inferring the volcano function for a user, where $K = 2$.

Specifically, we propose a heavy-tailed function to predict whether a distance is out-of-town or not for a user. Formally, for cluster k of the user:

$$V_k(d) = \frac{\alpha_k \beta_k}{s_k}\Big[\frac{\omega(d-b_k)}{s_k}\Big]\exp\Big[-\big(\frac{\omega(d-b_k)}{s_k}\big)^{-\alpha_k}\Big] \quad (4)$$

where $\omega(d) = \log(1+e^d)$ denotes the softplus function, $\beta_k \in \mathbb{R}$ is the zoom parameter to normalize $v_k(d)$ into $[0,1]$. $\alpha_k \in \mathbb{R}$ is the shape parameter which controls the shapes such as the width of the curve. $s_k \in \mathbb{R}$ and $b_k \in \mathbb{R}$ are the scale and bias parameters respectively. This function takes inspiration from Fréchet distribution, which is widely used in fitting heavy-tailed data [34]. As shown in Fig. 3(c), the curve starts sharply and falls slowly. If we remap the distances to geographical coordinates, this function looks like a volcano around the cluster. Therefore we call this function as *volcano function*.

Due to the clustering phenomenon, we can assign multiple personalized volcanoes for each user, whose shape could vary from person to person. For the sake of simplicity, we also omit the subscript u. Based on volcano functions, we can predict out-of-town probability of a distance to improve the performance on recommending remote venues. Then the remaining problem is, how to learn the proposed volcano functions for each user?

B. Inference for Volcano Function

As for the inference, we propose to sample pairs of (d_{ik}, y_{ik}) to train the proposed function, where d_{ik} is the distance and $y_{ik} \in \{0, 1\}$ is the indicator whether this distance is out-of-town or not. Based on these samples, we can use the classification loss function to train the function,

$$\begin{aligned}\ell(\alpha_k, \beta_k, s_k, b_k) = \sum_i & - y_{ik} \log V_k(d_{ik}) \\ & - (1-y_{ik})\log(1 - V_k(d_{ik}))\end{aligned} \quad (5)$$

However, there exists a major concern: how to label d_{ik} in the dataset for learning the function? For this problem, we propose to use the following sampling strategy:

- **Nearby Places** ($y_{ik} = 0$) **:** If the probability from the exponential distribution of a distance is larger than an extremely large threshold ξ_1 (e.g. 0.7), which means it is near the center of a cluster. We label it as nearby place.
- **Far Places** ($y_{ik} = 0$) **:** If the probability of a distance from the exponential distribution is smaller than an extremely small threshold ξ_2 (e.g. $1e-2$), which means it is too far from the user. We label it as a far place.
- **Out-of-Town Places** ($y_{ik} = 1$) **:** For distances with probability between ξ_2 and ξ_1, if the probability of the distance from exponential distribution is lower than the true tail distribution, we label it as an out-of-town place.

Here we use the facts that visited distances are heavy-tailed data and a light-tailed distribution is unable to model them. Then the problem is how to represent the true tail distribution. In order to address this issue, we introduce extreme value theory again, where the approximated cumulative density function can be written as,

$$1 - F(d) \approx (1 - F(t_k))\Big[1 - H_{\gamma_k}\Big(\frac{d - t_k}{\eta_k}\Big)\Big], d > t_k, \quad (6)$$

where $\eta_k > 0$, t_k is a large threshold and H_γ is defined as,

$$H_\gamma(d) = -(1 + \gamma_k \cdot d)^{-1/\gamma_k}, \quad (7)$$

where γ_k is called the extreme value index. For the parameters in the tail distribution, we propose to conduct the estimation as follows. First we define the threshold t_k as the T-largest distance in the cluster k. As for the extreme value index γ_k, we use Pickands Estimator [35] to do the parameter inference. For the scale parameter η_k, we estimate it with the methods proposed by [33]. The detailed inference is listed in the appendix. Based on the fitted tail distribution, we can sample some d_{ik} from $\tilde{\mathcal{L}}_{uk}$ and label it as y_{ik} for training the volcano function.

C. Regularized Learning Strategy

However, we encounter a new problem during the inference, that is, the optimization hardly converges. We find the main reason is that the expectation and momentum of the function does not exists due to the heavy tail. Therefore we will encounter gradients explosion or vanish during the inference. In order to learn the parameters effectively, we propose to add several penalty terms on the classification loss to ensure the convergence. Specifically, we consider the following statistics of volcano function:

- Although the expectation of the function may not exist, the median of the gate however have a closed-form formula as $(\ln 2)^{-\frac{1}{\alpha_k}}$. By denoting the median of positive samples is ψ_{uk}, then we propose the first regularization term
$$\ell_1(\alpha_k) = \|(\ln 2)^{-\frac{1}{\alpha_k}} - \psi_k\|^2, \tag{8}$$
where the approximation $\omega(x) \approx x$ is applied.
- We also add a penalty on s_k by regularizing the mode of the function (i.e., the value for which the function takes its maxima) to the median:
$$\ell_2(\alpha_k, s_k, b_k) = \|b_k + s_k\Big(\frac{\alpha_k}{1+\alpha_k}\Big)^{\frac{1}{\alpha_k}} - \psi_k\|^2$$
- In order to normalize the output from volcano function into range $[0, 1]$, we regularize the gate's maximum to 1 by
$$\ell_3(\alpha_k, \beta_k) = \|\frac{\alpha_k+1}{s_k}\Big(\frac{\alpha_k+1}{\alpha_k}\Big)^{\frac{1}{\alpha_k}} e^{-\frac{\alpha_k+1}{\alpha_k}} - \frac{1}{\beta_k}\|^2$$
- Finally we regularize bias b by $\ell_4(b_k) = \|b_k\|^2$

By adding these penalty terms to the original binary cross-entropy loss, we are therefore able to infer the parameters by gradient-based methods such as Adam [36]. After this stage, we have finally finished the learning of volcano function. The whole algorithm is listed in Alg. 1.

D. Summary

As we have discussed above, current methods lack the ability of modeling out-of-town distances because observations follow heavy-tailed distribution. In this section, we propose to use prior information. Specifically, we consider the factors of *neither close nor too far* property and heavy-tailed data, and propose a new function called *volcano function*. Furthermore, we present a regularized learning strategy to infer parameters in the function. As is depicted in Fig. 3, it is easy to see that the function can indeed measure the out-of-town distances well. Interestingly, our proposed method also captures the differences between different clusters. For instance, the user has a tendency to visit more remote venues in cluster 1 than those in cluster 2, which may correspond to the real world scenario where, e.g., cluster 1 may have more convenient transportation than cluster 2 to encourage user's exploration. Therefore the volcano function for cluster 1 is wider than cluster 2. In summary, we tackle two aforementioned challenges by:

Algorithm 1 Inference of volcano function for a user.

```
Input: Visited Locations 𝓛_u by user u.
Initialize: Parameters α_k, β_k, s_k, b_k
Infer μ_k, Σ_k, ρ_ik by detecting clusters          (Eq. 13)
for each cluster k do
    Form filtered POIs 𝓛̃_uk by ρ_ik, ξ              (Eq. 2)
    Fit exponential distribution with parameter λ_k, m_k
    Set t_k as the T-largest distance in 𝓛̃_uk
    Infer γ_k by Pickands Estimator
    Infer η_k by the method in [33]
    Establish tail distribution with parameters t_k, γ_k, η_k
    Sample (d_ik, y_ik) from 𝓛̃_uk                   (Sec. IV-B)
    Learn α_k, β_k, s_k, b_k in volcano function by:
end for
```

$$\min_{\alpha_k,\beta_k,s_k,b_k} \ell(\alpha_k, \beta_k, s_k, b_k) + \ell_1(\alpha_k) + \ell_2(\alpha_k, s_k, b_k) + \ell_3(\alpha_k, \beta_k) + \ell_4(b_k) \tag{9}$$

- **Heavy-tailed Data:** We propose a novel volcano function to address this issue. Compared with previous light-tailed distributions, our function will not under-estimate the probability of out-of-town distances.
- **Limited Observations:** We introduce the prior knowledge to this problem, for example, sampling pairs (d_{ik}, y_{ik}) and the proposed regularized learning. As we can see from the results, we can model out-of-town distances effectively with only few samples.

V. Recommendation Model

As we can see from the results, we can learn effective personalized volcanoes for each user. In this section we will present how to incorporate learned volcanoes into location recommendation.

A. Volcano Network

We further implement our proposed function as a novel network structure called *Volcano Network* (VolNet), as is shown in Fig. 4. Given a user u and a venue i, the network first outputs two vectors: *peak vector* $P_{ui} \in \mathbb{R}^K$ and *volcano vector* $V_{ui} \in \mathbb{R}^K$. The volcano vector describes whether the venue is out-of-town for this user or not, where peak vector measures how close is the venue to the user. Overally, our proposed VolNet consists of five parts:

- **Distance Gate**: This gate transforms the representation of longitude and latitude ϕ_i to d_{uik}, which is the distance between the venues and the learned clusters, as is defined in Eq. **??**. The pre-trained parameters in this gate are the center of each cluster μ_{uk} and the covariance of each cluster Σ_{uk} in Eq. 13.
- **Peak Function**: This gate is the function to calculate peak vector P_{ui}. For cluster k,
$$P_{uik} = \exp(-(d_{uik} - m_{uk})/\lambda_{uk}), \tag{10}$$

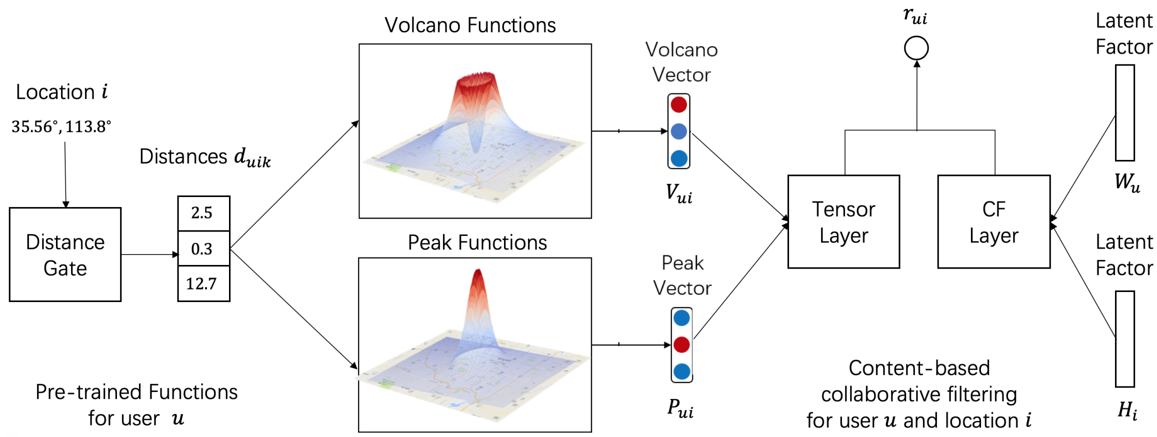


Fig. 4. The overview of volcano network. Given a pair of user u and location i, VolNet will output the predicting value r_{ui}.

where m_{uk} and λ_{uk} are parameters in the estimated exponential distribution.

- **Volcano Function**: This gate calculates the volcano vector V_{ui} by proposed function in Eq. 4: $V_{uik} = V_{uk}(d_{uik})$, where $\alpha_{uk}, \beta_{uk}, s_{uk}$ and b_{uk} are the leanred parameters of the volcano function.
- **Tensor Layer**: This layer calculates the content-based rating between user u and location i by,

$$\hat{r}_{ui} = X_u^T P_{ui} + Y_u^T V_{ui} + P_{ui}^T M_u V_{ui}, \quad (11)$$

where $X_u, Y_u \in \mathbb{R}^K$ are the weight on peak vector and volcano vector respectively, which determines the personalized preference on whether to choose distant venues or not. The purpose of introducing $M_u \in \mathbb{R}^{K\times K}$ here is that this form helps find latent correlations between the peak vector and volcano vector [37]. In our framework, it describes the personalized correlation between visiting nearby places and visiting out-of-town venues.
- **CF Layer**: This layer calculates the recommending rating of location i by $r_{ui} = \hat{r}_{ui} + W_u^T H_i$, where $W_u, H_i \in \mathbb{R}^D$ are latent factors in collaborative filtering.

B. Inference

Considering that the peak vector P_{ui} and the volcano vector V_{ui} are calculated from the pre-trained volcano functions, the rest of the parameters can be inferred by gradient-based methods such as Adam [36] under the following ranking loss [6],

$$\max_{W,H,X,Y,M} \sum_{u\in\mathcal{U}} \sum_{i\in\mathcal{L}_u, i'\notin\mathcal{L}_u} \sigma(r_{ui} - r_{ui'}), \quad (12)$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function. The goal is to maximize the rating of visited venues compared with the unvisited. We also deliver a concise analysis on the time complexity of our model. The whole training comprises the following stages, that is, for each user u,

- $\boldsymbol{\mu}, \boldsymbol{\Sigma}$: We use fuzzy c-means algorithm for the optimization, whose complexity is $O(|\mathcal{L}_u|K^2)$.
- m, λ: They are inferred by doing statistics on the original data, whose complexity is linearity $O(|\mathcal{L}_u|)$

TABLE II
DATASETS DESCRIPTION.

	Foursquare	Yelp	Meetup
Users	48,943	2,293	30,886
POIs	332,971	30,886	18,994
Check-ins	1,065,553	573,703	860,888
Max Longitude	$-70.01°$	$139.91°$	$-3.11°$
Min Longitude	$-124.65°$	$139.47°$	$-115.37°$
Max Latitude	$48.99°$	$35.86°$	$55.98°$
Min Latitude	$25.12°$	$35.51°$	$32.87°$

- Volcano function: The complexity of fitting tail distribution is $O(|\mathcal{L}_u|T)$. Meanwhile, the sampling and the training complexity is $O(|\mathcal{L}_u|G)$, where G is the sampling size for training volcano functions.
- CF model: The complexity is $O(|\mathcal{L}_u|(D + K^2))$.

Therefore for each user, the total time complexity is $O(|\mathcal{L}_u|(K^2+G+D+T))$, which is the constant times to the size of visited POIs.

VI. Empirical Results

In this section we report our empirical results mainly from two aspects, accuracy and serendipity in POI recommendation. In particular, the research questions are:

- **RQ1:** Could we enhance the performance of location recommendation compared with the state-of-the-art methods?
- **RQ2:** Is the improvement caused by our modeling on out-of-town venues?
- **RQ3:** What is the influence of hyper-parameters?

A. Experimental Settings

1) Dataset & Metrics: We choose three standard check-in datasets of different size and spatial range: Foursquare[1], Yelp[2] and Meetup[3]. Their statistics are listed in Table II. By conducting 10-fold cross validation in each experiment, we report the averaged results with corresponding p-values calculated by paired t-test. For the evaluation, we use $Precision@N$, $Recall@N$, $F1@N$ and Area Under the ROC curve (AUC) as measurement. We also validate the results under different N.

[1]https://foursquare.com/
[2]https://www.yelp.com/
[3]https://www.meetup.com/

TABLE III
PERFORMANCE IN TERMS OF PRECISION@N, RECALL@N AND F1@N AMONG VARIOUS APPROACHES ON DIVERSE DATASETS, WHERE THE BEST RESULTS ARE HIGHLIGHTED IN BOLD. * DENOTES A SIGNIFICANT DIFFERENCE COMPARED TO THE BEST RESULT, ACCORDING TO THE PAIRED T-TEST FOR $p < 0.05$.

N	Model	Foursquare			Yelp			Meetup		
		Precision@N	Recall@N	F1@N	Precision@N	Recall@N	F1@N	Precision@N	Recall@N	F1@N
3	NMF	0.2840*	0.0280*	0.0510*	0.3218*	0.0026*	0.0052*	0.2530*	0.0897*	0.1324*
	MGM	0.2980*	0.0587*	0.0981*	0.3720*	0.0152*	0.0272*	0.3415*	0.1020*	0.1570*
	GeoMF++	0.3389	0.1803*	0.2354*	**0.4662**	0.0312*	0.0584*	0.3899*	0.1425*	0.2087*
	GeoCNTN	0.3422	0.1699*	0.2271*	0.4425	0.0479*	0.0864*	0.3715*	0.1209*	0.1824*
	VolNet	**0.3450**	**0.2960**	**0.3186**	0.4378*	**0.1712**	**0.2462**	**0.4190**	**0.2576**	**0.3190**
8	NMF	0.2573*	0.1573*	0.1952*	0.3184*	0.0209*	0.0393*	0.2452*	0.1274*	0.1677*
	MGM	0.2667*	0.2217*	0.2421*	0.3538*	0.0592*	0.1014*	0.3231*	0.1357*	0.1912*
	GeoMF++	0.2877*	0.3120*	0.2994*	**0.4380**	0.0821*	0.1383*	0.3561*	0.1839*	0.2425*
	GeoCNTN	0.3020*	0.2977*	0.2998*	0.4295	0.1028*	0.1659*	0.3428*	0.1637*	0.2216*
	VolNet	**0.3508**	**0.3712**	**0.3607**	0.4250	**0.2275**	**0.2963**	**0.4033**	**0.2919**	**0.3386**
15	NMF	0.1980*	0.3022*	0.2392*	0.2681*	0.1419*	0.1856*	0.2387*	0.2178*	0.2278*
	MGM	0.2180*	0.3793*	0.2769*	0.2960*	0.1528*	0.2016*	0.2850*	0.2324*	0.2560*
	GeoMF++	0.2392*	0.4113*	0.3025*	**0.3762**	0.2089*	0.2686*	0.3231*	0.2466*	0.2797*
	GeoCNTN	0.2415	0.3824*	0.2960*	0.3718	0.1997*	0.2598*	0.3081*	0.2502*	0.2761*
	VolNet	**0.2582**	**0.4391**	**0.3252**	0.3669	**0.3103**	**0.3362**	**0.3863**	**0.3497**	**0.3671**

2) Baselines: We compare our proposed VolNet with the following state-of-the-art models for POI recommendation.

- NMF: Non-Negative Matrix Factorization (NMF) [38], which is an effective collaborative filtering model in POI recommendation [2], [20]. NMF works by predicting user-POI matrix directly without using any geographical information.
- MGM: Multi-center Gaussian Model [18], which combines matrix factorization with probability of a POI's belonging to pre-trained user's clusters.
- GeoMF++: Geographical Matrix Factorization [2], which combines non-parametric kernel density estimator of user's visited geographical locations into matrix factorization and is able to model distant locations compared with MGM. We use GeoMF++ [8] as baseline, which is the improved version of GeoMF.
- GeoCNTN: Geographical Convolutional Neural Tensor Network (GeoCNTN) [7] proposes to use a hybrid network structure to model both global and local views of visited locations for entities in LBSN. In our experiments, we utilize the global view only because each venue simply occupies one location.

3) Other details: All the latent factor models have the same dimension of embedding as 20. The learning rate for Adam optimizer is set as 0.001, the parameters are initialized by Gaussian initializer with mean 0 and standard deviation 0.01, and all the regularization coefficients are set as 0.1.The cluster number $K = 5$, and we also validate influence of different clusters. For the other hyper-parameters, we use the validation results to determine the best value, for instance, $T = 10$ in learning heavy-tailed distribution, and $G = 40$, $\xi = 0.5, \xi_1 = 0.8$ and $\xi_2 = 1e - 3$ in learning the volcano function.

B. Experimental Results

For RQ1, we compare the accuracy of our model with baselines (Table III). First, we notice geographical information is indeed important in POI recommendation from the comparison between NMF and other models. Especially in Yelp, the recall of NMF is close to 0 when recommending POIs. The comparison between MGM and GeoMF++ reassures our consideration that remote locations matter, e.g. the recall of GeoMF++ improves by about 10% compared with MGM on Foursquare dataset. Although such an improvement from GeoMF++ and GeoCNTN are observed compared with MGM, it is still not satisfying. As a strong empirical validation of our proposed VolNet, its performance is observed to be over the best baseline GeoMF++ by 14.6% in recall metric on Foursquare and by 20% in F1 score on Yelp. In addition, our model outperforms all other baselines in F1 score, which further validate the improved accuracy of VolNet. We also validate the effectiveness of the POI recommendations in AUC metric, as shown in Table IV. The results are similar to F1-score. For example, the results of MGM, GeoMF++ and GeoCNTN are observed to be much better than NMF, while all of them are poorer than our model. Especially in Yelp dataset, our model significantly outperforms the best baseline by 8.9% in AUC.

TABLE IV
AUC RESULTS FOR POI RECOMMENDATION.

	Foursquare	Yelp	Meetup
NMF	0.7868*	0.6369*	0.7827*
MGM	0.8273*	0.7600*	0.8309*
GeoMF++	0.8326*	0.7842*	0.8330*
GeoCNTN	0.8412*	0.7795*	0.8219*
VolNet	**0.8742**	**0.8425**	**0.8721**

As we can see from the previous results, we largely improve the recall, which means that we can find more potential POIs. Then the problem is, whether the improvement is caused by our modeling on out-of-town venues or not. In order to validate this, we pay attention to recommendation on places that are far from users. Specifically, we use Eq. 3 to filter close venues for each user to form a subset of the test set. According to previous researches that users often visit 30% far places [15], we keep 30% venues with the above approach. Due to the lack of space and the fact that the performance of GeoCNTN is similar to GeoMF++, we only plot GeoMF++ here for demonstration. As we can see from the results, VolNet largely outperforms all the baselines in recall on each dataset, which proves that VolNet indeed recommends far places to improve the performance. On the other side, our methods also outperforms all baselines in precision, which proves that we can correctly measure whether

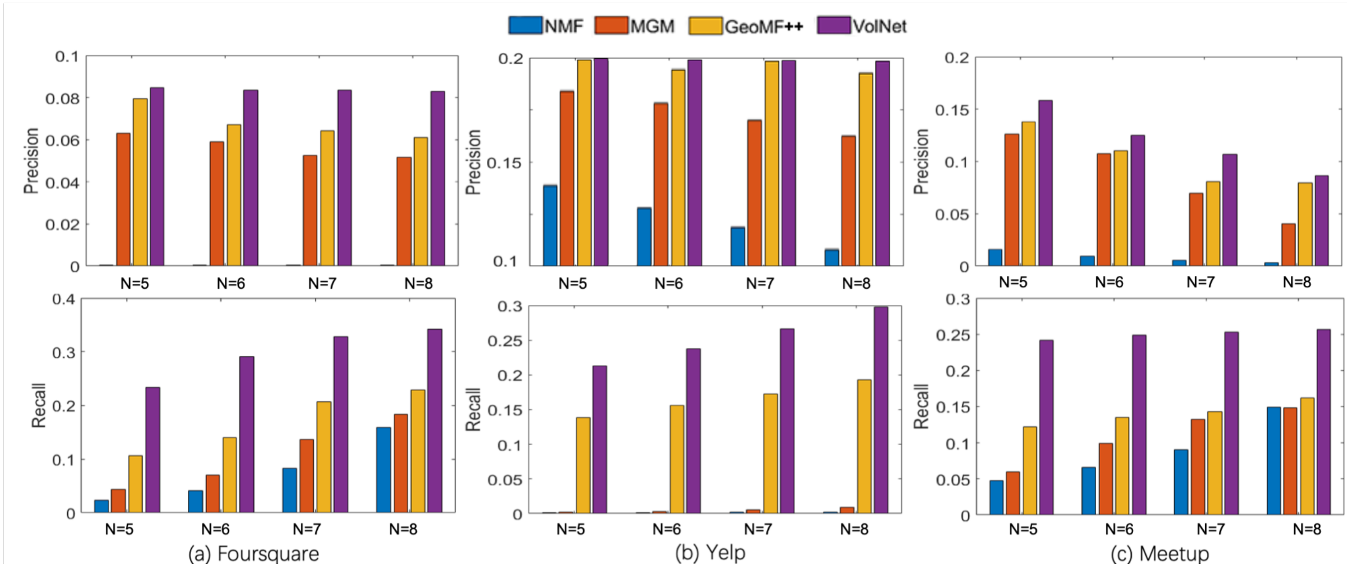


Fig. 5. Performance of models in recommending far venues, measured by $Precision@N$ and $Recall@N$.

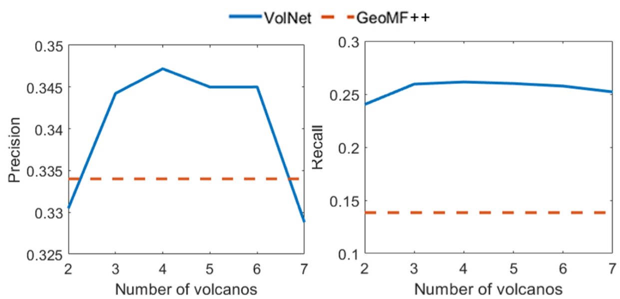


Fig. 6. Experimental results for investigating the influence of number of volcano. The results of Foursquare are reported on the top two figures, where Meetup's results are on the bottom.

the place is too far for the user to reach.

Noticeably, the spatial range of the dataset also influences the accuracy to some degree. In large datasets, the visited venues are more centered for each user on the huge map. On the other side, venues in small datasets appear to be more scattered such as Yelp, which leads to the fact that visited distances are more heavy-tailed. As a consequence, it is much harder for recommendation models to predict correct remote POIs, e.g., the corresponding recall value is much smaller than those in the other datasets. Considering that VolNet tries to predict venues with out-of-town distances, the recall is much larger than all the baselines. For instance, we improve the recall from 0.2 to 0.31 when $N = 15$. Besides, although GeoCNTN utilizes deep neural network to extract geographical patterns from visited data, the performance of GeoCNTN is still similar to GeoMF++.

For RQ3, as is mentioned, we set the number of clusters as 5 for all the datasets. We expect a different number of clusters may probably influence the performance of recommendations. We validate this hypothesis with results presented in Fig. 6. We set the range of K from 2 to 7 on different datasets and compare the result with the best baseline GeoMF++ (in dash line). As we can see, the curves on change of performance roughly achieve its optimum around $K = 5$, with the exact optimal setting of K varying on different datasets slightly. For instance, as users in Yelp dataset visit more locations in average, the optimal K was observed to be 6, while in Meetup dataset the optimal value for K was around 4. Furthermore, the number of volcanoes is observed to have a larger influence on the precision metric than that on recall. We infer the reason mainly as, although more volcanoes can help user find more out-of-town venues, the locations near the cluster center would otherwise be less recommended. Therefore there is a trade-off between nearby and out-of-town places.

VII. Conclusion

In this paper, we focus on modeling personalized geographical preferences on out-of-town venues and propose a novel framework called Volcano Network (VolNet). Specifically we assume that those out-of-town venues should be neither too close or too far, and propose a new function called volcano function to model the personalized out-of-town distances. According to the empirical results, VolNet is validated to noticeably improve recall compared with the state-of-the-art methods. Indeed, there are many potential directions based on our work. For example, it is promising to utilize 2-dimensional geographical patterns rather than distances to study this problem. Furthermore, we will devise more functions for modeling out-of-town distances and compare the difference with our current choice. Finally, it would be interesting to carry out spatial behavior analysis through the lens of interpretable features obtained by VolNet.

Acknowledgment

We would like to thank the anonymous reviewers for their constructive comments. This work was supported in part by National Natural Science Foundation of China (61972099, U1636204, U1836213, U1836210, U1736208, 61972372), and

Natural Science Foundation of Shanghai (19ZR1404800). Min Yang is a faculty of Shanghai Institute of Intelligent Electronics & Systems, Shanghai Institute for Advanced Communication and Data Science, and Engineering Research Center of CyberSecurity Auditing and Monitoring, Ministry of Education, China.

Appendix

A. Fitting Exponential Distribution

At the first stage we use fuzzy C-Means [32] to find clusters of a user and train μ, Σ, which has been previously applied in LBSN tasks (e.g. [7]). The optimization objective of fuzzy CMeans is formulated as,

$$\min_{p_k,\mu_k,\Sigma_k} \sum_{i\in\mathcal{L}_u}\sum_{k=1}^{K} \rho_{ik}^{\epsilon}\cdot d_{ik},\ \ s.t.\ \sum_{k=1}^{K}\rho_{ik}=1 \tag{13}$$

where $d_{ik} = (\phi_i - \mu_k)^T\Sigma_k^{-1}(\phi_i-\mu_k)$, ρ_{ik} represents the probability of location i belonging to cluster k, with $\epsilon \in (1, +\infty]$ the factor of fuzziness. Several mature algorithms by iterative optimization can be efficiently applied to solve the optimization problem above [31]. Here we choose the method described in [39], with one iterative step in their algorithm as follows.

$$\begin{aligned} \mu_k &= \Big[\sum_{i\in\mathcal{L}_u}\rho_{ik}^{\epsilon}\Big]^{-1}\sum_i \big[\rho_{ik}\big]^{\epsilon}\phi_i \\ \rho_{ik} &= \left[\sum_{j=1}^{K}\Big[\frac{(\phi_i-\mu_k)^T\Sigma_k^{-1}(\phi_i-\mu_k)}{(\phi_i-\mu_j)^T\Sigma_j^{-1}(\phi_i-\mu_j)}\Big]^{\frac{1}{\epsilon-1}}\right]^{-1} \\ \Lambda_k &= \left[\sum_{j=1}^{K}\sum_{i\in\mathcal{L}_u}\rho_{ij}^{\epsilon}\right]^{-1}\sum_{j=1}^{K}\sum_{i\in\mathcal{L}_u}\rho_{ij}^{\epsilon}(\phi_i-\mu_k)(\phi_i-\mu_k)^T \end{aligned} \tag{14}$$

, where $|\cdot|$ is the determinant of a matrix and $\Sigma_k = |\Lambda_k|^{\frac{1}{2}}\Lambda_k^{-1}$

Based on pre-trained μ_k, Σ_k and the filtered cluster $\tilde{\mathcal{L}}_{uk}$, we further infer parameters in the exponential distribution, i.e. m_k, λ_k. Considering that m_k is the locale parameter and λ_k is the scale parameter, we propose to maximize the following exponential likelihood, where distances from locations in $\tilde{\mathcal{L}}_{uk}$ to the cluster center serve as observations. Formally, it is

$$\max_{m_k,\lambda_k}\sum_{k=1}^{K}\sum_{i\in\tilde{\mathcal{L}}_{uk}}\frac{1}{\lambda_k}\exp\big[-\frac{d_{ik}-m_k}{\lambda_k}\big] \tag{15}$$

, which can be easily solved with standard gradient descent algorithms.

B. Fitting Tail Distribution

As we have discussed above, common distributions such as Gaussian, Poisson and Exponential are light-tailed distributions [40]. On the other side, most data in real world applications such as climate data follow heavy-tailed distribution [41], [42], which have infinity momentum. As the formal definition, suppose variables $x > 0$, if its cumulative density function (c.d.f) F satisfies [43] $\int_0^{\infty} e^{tx}dF(x) = \infty\ \ \forall t > 0$, then we call the distribution of x is a *heavy-tailed* distribution. In order to model the tail distribution of these data, extreme value theory propose to study the maximum of the observations. Generally speaking, suppose n random variables $X_1, \ldots, X_n$ are i.i.d sampled from an arbitrary distribution $F(x)$, then the distribution of the maximum should be 0:

$$\lim_{n\to\infty} P(\max(X_1,\cdots,X_n)\le x) = \lim_{n\to\infty}F^n(x) = 0 \tag{16}$$

However, previous works find that, if we conduct a linear transformation on $M_n = \max(x_1, \cdots, x_n)$, the distribution of the maximum will not degenerate:

Theorem 1 ([44], [45]). *If there exists sequences $(a_n)_{n\in\mathbb{Z}^+}, (\delta_n)_{n\in\mathbb{Z}^+}$ such that $\forall n \in \mathbb{Z}^+, a_n > 0$ and*

$$\lim_{n\to\infty} P\Big(\frac{M_n-\delta_n}{a_n}\le x\Big) = G(x) \ne 0 \tag{17}$$

Then the class of $G(x)$ must be of the following form

$$G_\gamma(x) = \begin{cases} exp\big[-(1+\gamma x)^{-1/\gamma}\big], 1+\gamma x>0, \gamma\ne 0 \\ exp\big[-e^{-x}\big], \gamma = 0\end{cases} \tag{18}$$

, where γ is called extreme value index. Besides, there exists a positive function a such that

$$\lim_{t\to x^*} P\Big(\frac{X-t}{a(t)} > x|X>t\Big) = H_\gamma(x) \tag{19}$$

, where $x^* = \sup\{x : F(x) < 1\}$ and $H_\gamma(x) = \log G(x)$.

In the theorem above, $G_\gamma(x)$ is conventionally called the class of extreme distributions, which describes the limiting probability of maximum of parent distribution $F(x)$. In consideration of the correlation between parent distribution $F(x)$ and limiting distribution $G_\gamma(x)$ [45], this theorem can be regarded as the law of large numbers for maximum to some extent [46]. In this paper, we propose to use this theorem to approximate the tail distribution:

$$1-F(d)\approx(1-F(t_k))\Big[1-H_{\gamma_k}\Big(\frac{d-t_k}{\eta_k}\Big)\Big], d>t_k \tag{20}$$

Then for each cluster k, we do inference on parameters as follows:

- **Threshold:** We choose the threshold $t_k = d_k^{(T)}$, where $d_k^{(T)}$ represents the T-largest distance in cluster k. Thus the probability of $1 - F(t_k)$ can be approximated as $T/|\mathcal{L}_{uk}|$
- **Extreme Value Index:** We use the Pickands Estimator to estimate γ_k, which is a mature algorithm for estimating γ_k and is guaranteed to converge [35]:

$$\gamma_k = \frac{1}{\ln 2}\log\frac{d_k^{(T)}-d_k^{(2T)}}{d_k^{(2T)}-d_k^{(4T)}} \tag{21}$$

- **Scale Parameter:** We estimate the term a_k by methods proposed by [33]:

$$\begin{aligned} a_k &= \tfrac{1}{2}d_k^{(T)}\Theta_k\Big(1-\tfrac{\Theta_k^2}{\hat{\Theta}_k}\Big)^{-1} \\ \Theta_k &= \frac{1}{T}\sum_{\tau=0}^{T-1}\log d_k^{(\tau)} - \log d_k^{(T)} \\ \hat{\Theta}_k &= \frac{1}{T}\sum_{\tau=0}^{T-1}\big[\log d_k^{(\tau)}-\log d_k^{(T)}\big]^2 \end{aligned} \tag{22}$$

References

[1] W. Wang, H. Yin, L. Chen, Y. Sun, S. Sadiq, and X. Zhou, "Geo-sage: A geographical sparse additive generative model for spatial item recommendation," in *KDD*, 2015.

[2] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation," in *SIGKDD*, 2014.

[3] M. Ye, P. Yin, W. C. Lee, and D. L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," pp. 325–334, 2011.

[4] J.-D. Zhang and C.-Y. Chow, "Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations," in *SIGIR*, 2015, pp. 443–452.

[5] M. Aliannejadi and F. Crestani, "Personalized context-aware point of interest recommendation," *ACM Trans. Inf. Syst.*, vol. 36, no. 4, pp. 45:1–45:28, Oct. 2018. [Online]. Available: http://doi.acm.org/10.1145/3231933

[6] X. Li, G. Cong, X. L. Li, T. A. N. Pham, and S. Krishnaswamy, "Rank-geofm:a ranking based geographical factorization method for point of interest recommendation," 2015, pp. 433–442.

[7] D. Ding, M. Zhang, X. Pan, D. Wu, and P. Pu, "Geographical feature extraction for entities in location-based social networks," in *WWW*, 2018.

[8] D. Lian, K. Zheng, Y. Ge, L. Cao, E. Chen, and X. Xie, "Geomf++: Scalable location recommendation via joint geographical modeling and matrix factorization," *ACM Trans. Inf. Syst.*, vol. 36, no. 3, pp. 33:1–33:29, Mar. 2018. [Online]. Available: http://doi.acm.org/10.1145/3182166

[9] J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel, "Recommendations in location-based social networks: a survey," *GeoInformatica*, vol. 19, no. 3, pp. 525–565, 2015.

[10] Y. Yu and X. Chen, "A survey of point-of-interest recommendation in location-based social networks," in *Workshops at the AAAI 2015*.

[11] W. R. Tobler, "A computer movie simulating urban growth in the detroit region," *Economic Geography*, vol. 46, no. sup1, pp. 234–240, 1970.

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

[13] G. Ference, M. Ye, and W.-C. Lee, "Location recommendation for out-of-town users in location-based social networks," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2013, pp. 721–726.

[14] S. Khoshahval, M. Farnaghi, M. Taleai, and A. Mansourian, *A Personalized Location-Based and Serendipity-Oriented Point of Interest Recommender Assistant Based on Behavioral Patterns*, 03 2018.

[15] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen, "Lcars:a location-content-aware recommender system," in *SIGKDD*, 2013.

[16] D. T. Ory and P. L. Mokhtarian, "When is getting there half the fun? modeling the liking for travel," *Transportation Research Part A: Policy and Practice*, vol. 39, no. 2-3, pp. 97–123, 2005.

[17] B. Liu, H. Xiong, S. Papadimitriou, Y. Fu, and Z. Yao, "A general geographical probabilistic factor model for point of interest recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1167–1179, 2015.

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

[19] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, "Time-aware point-of-interest recommendation," in *SIGIR*, 2013.

[20] J.-D. Zhang, C.-Y. Chow, and Y. Li, "igeorec: A personalized and efficient geographical location recommendation framework," *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 701–714, 2015.

[21] J.-D. Zhang and C.-Y. Chow, "Core: Exploiting the personalized influence of two-dimensional geographic coordinates for location recommendations," *Information Sciences*, vol. 293, pp. 163–181, 2015.

[22] T. Horozov, N. Narasimhan, and V. Vasudevan, "Using location for personalized poi recommendations in mobile environments," *Proc.int.symp.on Applications and the Internet*, pp. 124–129, 2006.

[23] M. Ye, P. Yin, and W. C. Lee, "Location recommendation for location-based social networks," in *ACM Sigspatial International Symposium on Advances in Geographic Information Systems, Acm-Gis 2010, November 3-5, 2010, San Jose, Ca, Usa, Proceedings*, 2010, pp. 458–461.

[24] S. Scellato, A. Noulas, and C. Mascolo, "Exploiting place features in link prediction on location-based social networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 1046–1054.

[25] Y. Liu, W. Wei, A. Sun, and C. Miao, "Exploiting geographical neighborhood characteristics for location recommendation," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 739–748.

[26] N. Zheng, Q. Li, S. Liao, and L. Zhang, "Flickr group recommendation based on tensor decomposition," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 737–738.

[27] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *SIGKDD*. ACM, 2011.

[28] 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 (TOIS)*, vol. 35, no. 4, p. 31, 2017.

[29] J. Manotumruksa, C. Macdonald, and I. Ounis, "A contextual attention recurrent architecture for context-aware venue recommendation," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018, pp. 555–564.

[30] T.-A. N. Pham, X. Li, and G. Cong, "A general model for out-of-town region recommendation," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 401–410.

[31] P. Wang, "Pattern recognition with fuzzy objective function algorithms (james c. bezdek)," *Siam Review*, vol. 25, no. 3, pp. 442–442, 1983.

[32] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," 1973.

[33] L. D. Haan and A. Ferreira, "Extreme value theory: an introduction," *Series in Operations Research and Financial Engineering*, vol. 60, no. 1, pp. 1–20, 2006.

[34] L. De Haan and A. Ferreira, *Extreme value theory: an introduction*. Springer Science & Business Media, 2007.

[35] J. P. Iii, "Statistical inference using extreme order statistics," *Annals of Statistics*, vol. 3, no. 1, pp. 119–131, 1975.

[36] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[37] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *NIPS*, 2013.

[38] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2001.

[39] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on*. IEEE, 1979, pp. 761–766.

[40] L. von Bortkiewicz, *Variationsbreite und mittlerer Fehler*. Berliner Mathematische Gesellschaft, 1921.

[41] T. Okubo and N. Narita, "On the distribution of extreme winds expected in japan," *National Bureau of Standards Special Publication*, vol. 560, p. 1, 1980.

[42] J. A. Tawn, "Estimating probabilities of extreme sea-levels," *Journal of the Royal Statistical Society*, vol. 41, no. 1, pp. 77–93, 1992.

[43] T. Rolski, H. Schmidli, V. Schmidt, and J. L. Teugels, *Stochastic processes for insurance and finance*. John Wiley & Sons, 2009, vol. 505.

[44] R. A. Fisher and L. H. C. Tippett, "Limiting forms of the frequency distribution of the largest or smallest member of a sample," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 24, no. 2, pp. 180–190, 1928.

[45] Gnedenko, "Sur la distribution limite du terme maximum d'une série aléatoire," *Annals of Mathematics*, vol. 44, no. 3, pp. 423–453, 1943.

[46] S. Kotz and S. Nadarajah, *Extreme value distributions. Theory and applications*. Prentice Hall,, 2000.