Lecture Notes in Social Networks

Tansel Özyer
Reda Alhajj *Editors*

Machine Learning Techniques for Online Social Networks

Springer

Lecture Notes in Social Networks

More information about this series at http://www.springer.com/series/8768

Tansel Özyer • Reda Alhajj
Editors

Machine Learning Techniques for Online Social Networks

Springer

Editors
Tansel Özyer
Department of Computer Engineering
TOBB University of Economics
and Technology
Ankara, Turkey

Reda Alhajj
Department of Computer Science
University of Calgary
Calgary, AB, Canada

ISSN 2190-5428 ISSN 2190-5436 (electronic)
Lecture Notes in Social Networks
ISBN 978-3-319-89931-2 ISBN 978-3-319-89932-9 (eBook)
https://doi.org/10.1007/978-3-319-89932-9

Library of Congress Control Number: 2018943402

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG part of Springer Nature.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Machine learning techniques are essential for social network analysis leading to effective and guided decision making. This book contains 11 chapters that focus on both machine learning techniques and social networks in link to a variety of applications. These chapters were thoroughly reviewed and comprehensively revised into the content of this book. We would like to thank the authors and reviewers as well as Springer Nature officers who worked hard to produce this book and make it available to the readers.

In the first chapter, authors deal with the problem of extracting functionally similar regions in urban streets in terms of spatial networks. They proposed an acceleration method of the functional cluster extraction (FCE) algorithm using the lazy evaluation and pivot pruning techniques. Following the first chapter, the work described in the second chapter is motivated on reducing the size of a graph to its core part. It is responsible for maximizing its delta hyperbolicity using the local dominance relationship between vertices. Delta hyperbolicity is used to give the metrical closeness of the structure of a graph to the structure of a tree. In the third chapter, authors develop a general-purpose benchmark for the evaluation of the resources linked to open social network applications in twofold. First, a dynamic workload is developed according to the main current online social network user's roles. Second, a complete framework is offered to be able to provide all classes of metrics for general-purpose performance. The fourth chapter addresses a stochastic dynamic programming model for the problem of impression dissemination. A heuristic method has been developed to approximate optimal solutions accurately and efficiently; they developed a method of exploiting communities in reciprocal online social network graphs by dividing larger instances of the problem into smaller. The fifth chapter proposes an approach to estimate the order of magnitude of pirated content to understand fundamental properties of popularity of torrents that are used to share pirated content. The sixth chapter presents an adaptive solution for privacy customization in OSNs by using deep reinforcement learning. Privacy labels are generated dynamically for OSN users based on trust on Twitter. The seventh chapter mentions the noise problem in the area of network analysis. It tries to figure out how community scoring functions and centrality measure parameters get affected by varying levels of noise. Its effects are discussed according to sensitivity, robustness, and reliability. The eighth chapter focuses on the extraction of useful data in terms of quality and time by improving search results. A framework has been proposed to eliminate the duplication, and then a clustering method is applied to filter and classify the results. The ninth chapter studied the dynamic network of relationships among avatars in the massively multiplayer online game Planet side 2. Two separate servers of this game have been merged, and its evolution has been observed. The tenth chapter presents a privacy preserving decentralized personal online social network platform that implements a cloud-backed peer-to-peer decentralized OSN using mobile devices. User's privacy is ensured by encryption. The 11th chapter performs emotion detection after extracting tweets on various topics. After utilizing natural language processing techniques, they were classified into 32 emotion classes. Emotions were then analyzed with respect to gender and location of the user and the time of the tweet.

Contents

Acceleration of Functional Cluster Extraction and Analysis of Cluster Affinity

Takayasu Fushimi, Kazumi Saito, Tetsuo Ikeda, and Kazuhiro Kazama

Abstract In this paper, we address the problem of extracting functionally similar regions in urban streets regarded as spatial networks. To efficiently deal with several large-scale networks, we propose a fast extraction method of functionally similar regions using the lazy evaluation and pivot pruning techniques. In our experiments using the urban streets of 12 cities from all over the world, compared with a state-of-the-art method based only on the lazy evaluation technique, we show that our proposed method achieved a reasonably high acceleration performance. We also show that our method could extract major functional clusters as regions corresponding to downtown, suburban, and mountainous areas for all the 12 spatial networks used in our experiments, and each cluster for the same area had quite similar characteristics in terms of the relations among the other clusters.

1 Introduction

Studies of the structures and functions of large complex networks have attracted a great deal of attention in many different fields, such as sociology, biology, physics, and computer science [1]. As a particular class, we focus on spatial networks embedded in real spaces, like urban streets, whose nodes occupy precise positions in two- or three-dimensional Euclidean space and whose links are actual physical connections [2]. In this paper, we concentrate on urban streets, treat them as large spatial networks, and address the problem of extracting functionally similar regions,

T. Fushimi (✉)
Tokyo University of Technology, Hachioji City, Tokyo, Japan

K. Saito · T. Ikeda
University of Shizuoka, Shizuoka City, Shizuoka, Japan
e-mail: k-saito@u-shizuoka-ken.ac.jp; t-ikeda@u-shizuoka-ken.ac.jp

K. Kazama
Wakayama University, Wakayama City, Wakayama, Japan
e-mail: kazama@ingrid.org

T. Özyer, R. Alhajj (eds.), *Machine Learning Techniques for Online Social Networks*,
Lecture Notes in Social Networks, https://doi.org/10.1007/978-3-319-89932-9_1

as functional clusters [3] from these networks. Typical examples of functional clusters are regions which individually cover downtown, suburban, and mountainous areas. Such characteristics of regions will play important roles for developing and planning city promotion, travel tours, and so on, as well as understanding and improving the usage of urban streets. In particular, we can expect that our method can give more reasonable explanations to some ambiguously discussed notions such as the boundaries among downtown, suburban, and mountainous areas.

Compared with a conventional issue of extracting communities from networks [4], our issue is the same in that the nodes in the networks are divided into several groups. However, it is significantly different from the conventional one because we focus on the functional properties of nodes that are derived from a network structure [3]. For instance, for social networks where each node corresponds to a person, we extract groups of persons in terms of similar positions and/or roles with respect to others. Each cluster must be a connected graph in conventional clustering methods. However, the nodes in each functional cluster are not necessarily connected. Note that even though our functional clustering techniques are applicable to a wide variety of networks, we focus on the spatial networks constructed by mapping the ends and the intersections of streets into nodes and the streets between the nodes into links. This is because based on general knowledge about familiar cities, we can intuitively understand the functional properties of nodes and interpret the resultant functional clusters.

To extract functional clusters, we employ our previous algorithm [5], called the Functional Cluster Extraction (FCE) method, which consists of two phases: the calculation of feature vectors, called functional vectors, through a random walk process, and clustering these vectors by the K-medoids method based on a greedy algorithm, where we need great computational cost for the subsequent clustering phase. More specifically, let N be the number of functional vectors, which equals the number of nodes in the network, and let S be the dimension of functional vectors, which equals the time steps of the random walk process. After calculating the pair-wise distance of these vectors with computational cost $O(N^2S)$, we run the K-medoids clustering phase with computational cost $O(KN^2)$ when we have enough space in our main memory for storing all of the $N(N-1)/2$ distances. However, when N functional vectors are too large and the space in the main memory is insufficient, we need to recalculate most of the $N(N-1)/2$ distances for all K greedy steps at the K-medoids clustering phase; it amounts to computational cost $O(KN^2S)$. Note that in our experiments below, the typical values for these variables are $K = 10$, $N = 100,000$, and $S = 10,000$, which causes a huge amount of computation time. In our previous study [5], we proposed an acceleration algorithm of the K-medoids method using the lazy evaluation and pivot pruning techniques to extract functional clusters from large-scale spatial networks. Furthermore, we confirmed that functional clusters share similar characteristics for all of the six cities used in our previous experiments [6].

In this paper, we extended our conference paper [5] for the following three points: (1) we proposed a method for analyzing the adjacency structure of functional clusters for each spatial network by using a heat map of the affinity matrix based on the number of nodes connecting to different clusters, (2) we also proposed another method for analyzing the similarity structure among several given spatial networks by using a dendrogram based on the cosine similarity between functional vectors of those networks, and (3) we more intensively evaluated our proposed methods by conducting further experiments where we used twelve spacial networks including additional six cities.

This paper is organized as follows. After explaining related work in Sect. 2, we describe the notion and extraction method of functional clusters in Sect. 3 and the details of the acceleration algorithm in Sect. 4. Then after explaining our experimental design in Sect. 5, we evaluate the computational performance of our algorithm in Sect. 6. After that, we evaluate the characteristics of the extracted functional clusters in Sect. 7, and their adjacent structure in Sect. 8. In Sect. 9, we discuss the similarity among networks based on the functional structure. Finally we conclude in Sect. 10. For easy reference, we summarize the notation in Table 1.

2 Related Work

As mentioned above, the structures and functions of large spatial networks have often been studied [2, 7–11]. From structural viewpoints, centrality measures have been widely used to analyze such networks [2, 10], especially by extending the conventional notions of centrality measures on simple networks into those of weighted networks [8, 9]. From functional viewpoints, traffic usage patterns in urban streets have been investigated [7, 11]. Unlike these previous studies, in this paper, we focus on extracting the functional clusters as the intrinsic properties of these spatial networks. by using our FCE (Functional Cluster Extraction) method. Note also that our study spontaneously combines structural and functional viewpoints in terms of functional clusters.

Table 1 Notation

Symbol	Description and definitions
V	Set of nodes
R	Set of medoids(representative nodes), $R \subset V$
P	Set of pivot nodes, $P \subset V$
$\mathbf{x}_u$	Functional vector of node u
N	Number of nodes, $N = \|V\|$
S	Dimension of functional vectors
K	Number of medoids (clusters), $K = \|R\|$
H	Number of pivots, $H = \|P\|$
$\rho(u, v)$	Cosine similarity between functional vectors of nodes u and v
$\mu(u; R)$	Maximum similarity of node u, $\mu(u; R) = \max_{r \in R}\{\rho(u, r)\}$
$d(u, v)$	Euclidean distance between functional vectors of nodes u and v
$f(R)$	Objective function of K-medoids clustering
$g(w, R)$	Marginal gain of objective function

As mentioned above, functional properties can be assumed to a wide variety of networks. Thus, in sociology, similar notions of node functions or roles have been studied as structural equivalence [12] and regular equivalence [13], together with their extraction algorithms. These notions focus on local structures like relationships with adjacent nodes. Functional vectors in the FCE method, however, reflect not only local structures but global ones through a random walk process. Recently, more advanced techniques for role discovery [14–19] have been widely investigated by assuming and utilizing a scale-free property of networks. However, since the maximum degree of nodes in spatial networks like urban streets is restricted to a relatively small number, we cannot straightforwardly apply these techniques to such spatial networks. In terms of discovering regions of different functions in a city, Yuan et al. [20] proposed a method to classify regions by using a topic model for human mobility and POIs. Since the method is based on movement history, it is difficult to apply to suburban areas where data may not be sufficiently obtained.

Studies of community extraction are another prominent branch of complex network analysis. As mentioned above, we previously employed a method of extracting functional clusters [3]. This is because representative methods for extracting communities as densely connected subnetworks, which include the Newman clustering method based on the modularity measure [4], cannot directly deal with such functional properties. Also, conventional notions of densely connected subnetworks such as k-core [21] and k-clique [22] cannot work for this purpose. We naturally anticipate that these representative methods suffer from an intrinsic limitation for extracting functional similar nodes. It might also be difficult to straightforwardly apply these conventional methods to spatial networks, because the maximum degree of nodes in each network is generally restricted to a relatively small number, that is, densely connected subnetworks, which are unlikely to appear in these networks.

Here, we should emphasize that our FCE method is potentially applicable to a wide range of complex networks including social networks constructed from relations of people and information networks constructed from citations of papers. Then, we can expect to obtain functional clusters such as groups of leaders from each community in social networks and those of outstanding papers from each field in information networks, where the nodes in each functional cluster are not necessarily connected, although each cluster must be a connected graph in conventional clustering methods. In this paper, even though our functional clustering techniques are applicable to a wide variety of networks, we focus on the spatial networks constructed by mapping the ends and the intersections of streets into nodes and the streets between the nodes into links. This is because on the basis of general knowledge about familiar cities, we can intuitively understand the functional properties of nodes and interpret the resultant functional clusters. Examples of such functional clusters might include parts of streets constructed in planned cities like lattices, those reflected by geographical restrictions like cul-de-sacs, and so on.

In this paper, we focus on the FCE method which employs the K-medoids clustering method for dividing all of the nodes into groups of functionally similar nodes by greedy maximization of the objective function. For clustering large-scale datasets, we employ representative sampling algorithms like [23, 24]. Another previous work [23] focused on the fractal structure of the dataset and extracted critical sized subsets that hold the entire dataset structure. However, since they computed approximated centers or clusters from a stochastically selected relatively small amount of objects, the accuracy of the results was not guaranteed.

As another branch or research, the acceleration of clustering, the Elkan algorithm [25] and the Hamerly algorithm [26] avoid redundant distance calculations in the K-means algorithm, which divides N objects into K clusters. Acceleration is caused by effectively using the distance of the lower and upper bounds, which are derived from the triangle inequality. Recently, the hybrid Elkan and Hamerly algorithm [27] treated the number of lower bounds as a variable parameter in the one to K range to best exploit the strength of each algorithm. By using pivots which efficiently select initial medoids and accelerate the convergence in iterative steps, Paterlini et al. [28] proposed a fast algorithm of K-medoids clustering. Unlike these existing methods, based on a greedy approach, it is guaranteed that the FCE method can produce a unique greedy solution with reasonably high quality because of submodularity of the objective function.

3 Functional Cluster

In this study, we extract functional clusters that consist of functionally similar nodes from a given network. Figure 1 shows an example of functional clusters extracted from a synthetic network of two web-like graphs connected by a single link, where the clusters extracted by our FCE method are distinguished by blue, red, and green. Functional clusters are formed by nodes at the center (blue), intermediate (red), and peripheral parts (green) for each web-like graph. Thus, for spatial networks constructed from urban streets, we expect to obtain such functional clusters as city centers with our designated resolution that is controlled by K clusters. Here, recall that our method is applicable to a wide range of networks including social networks constructed from human relations.

3.1 Functional Cluster Extraction Method

For extracting functional clusters, we revisit the FCE method [3], which consists of two steps: the calculation of functional vectors and clustering them. Let $G = (V, E)$ be a given spatial network, where $V = \{u, v, w, \cdots\}$ and $E = \{(u, v), \cdots\}$ stand for sets of nodes and links, respectively; we denote the number of nodes by $N = |V|$. In this paper, we only consider undirected networks such that $(u, v) \in E$ implies $(v, u) \in E$, but we can straightforwardly extend our approach to deal with directional networks. For each node $u \in V$, we denote the set of its adjacent nodes by $\Gamma(u) = \{v \mid (u, v) \in E\}$. By considering the following iterative process,

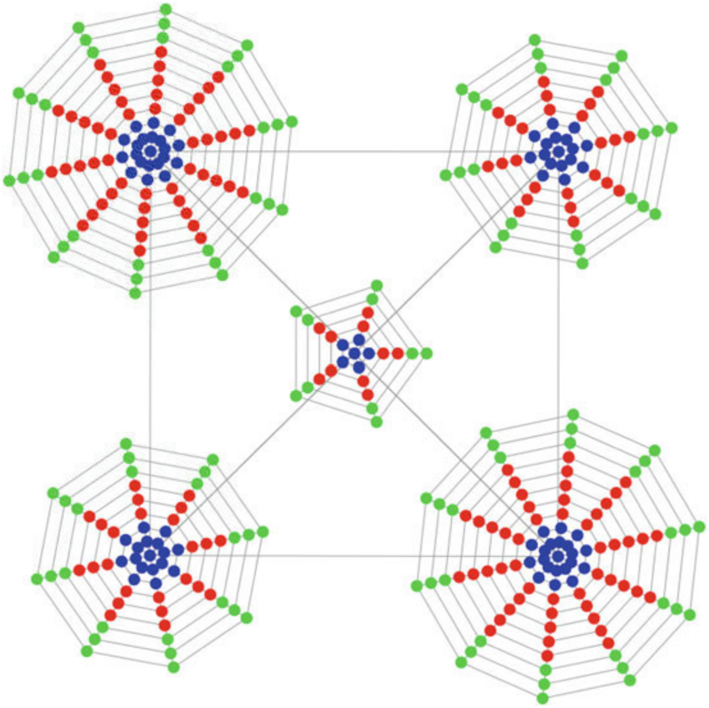


Fig. 1 Synthetic network like an urban street and its functional clusters

$$y_s(u) = \sum_{v \in \Gamma(u)} \frac{y_{s-1}(v)}{|\Gamma(v)|}, \tag{1}$$

we can define random walk probability $y_s(u)$ of node u at iteration step s, where $y_s(v) \geq 0$ and $\sum_{v \in V} y_s(v) = 1$. This model is basically a special version of PageRank where teleportation jump probability α is set to 0. Note that under a mild condition, $y_s(u)$ converges to a value proportional to the degree of node u, that is, $|\Gamma(u)|/\sum_{v \in V} |\Gamma(v)|$. We focus on the PageRank score vectors at each iteration step s, that is, $\{\mathbf{y}_0, \cdots, \mathbf{y}_S\}$, where we set the initial vector to $\mathbf{y}_0 = (1/N, \ldots, 1/N)$, and S stands for the final step of the iterations. Then for each node $u \in V$, an S-dimensional vector can be defined by

$$\mathbf{x}_u = (y_1(u), \cdots, y_S(u)), \tag{2}$$

where $y_s(u)$ also corresponds to the PageRank score of node u at iteration step s. Hereafter, $\mathbf{x}_u$ is the functional vector of node u. The functional vector of each node contains not only local information like the degree of the node as a converged value but also global information accumulated through a random walk process like PageRank. Thus, by clustering the functional vectors, we expect to extract groups of similar nodes in terms of positions and/or roles with respect to the other nodes. Note that we set dimension S of the functional vector to a relatively large value, that is, 10,000, because the diameters of the spatial networks in our experiments were generally large.

We employed the functional vector described above because we basically assume that the functional properties of such nodes as hierarchical levels, relative locations, and/or roles with respect to the other nodes are embedded into the network structure. On the other hand, the PageRank scores at each iteration step also reflect the network structure. Therefore, as an approximation, the functional properties are also represented by vector $\mathbf{x}_u$. This functional vector's virtue is that it depends not only on the degree of nodes but also on the local structure, such as how to connect with neighboring nodes, and global structures, such as which community the node belongs to. The beginning steps of the iteration especially reflect the local function of each node, and the whole shape of the functional vector represents the function of each node for the global structure. Thus, the dimensionality of functional vectors can be set relatively large.

Here, we also note two points in terms of calculating functional vectors from spatial networks. First, we set the teleportation jump probability to $\alpha = 0$ because this setting leads to a more natural random walk process over urban streets, although the final step of the iterations should be set to a reasonably large number, that is, $S = 10{,}000$, in order to capture global properties of nodes. In our preliminary experiments, we confirmed that the obtained results were substantially robust with respect to different settings of S if it is set to reasonably large ones. Second, the intuitive idea to explain the usefulness of using functional vectors is that the functionally similar areas are visited similarly during a random walk process starting from a uniform probability vector. In our experiments described later, we empirically show that our proposed method can produce a series of promising results.

Based on the following cosine similarity $\rho(u, v)$ between each pair of functional vectors, $\mathbf{x}_u$ and $\mathbf{x}_v$,

$$\rho(u, v) = \left\langle \frac{\mathbf{x}_u}{\|\mathbf{x}_u\|}, \frac{\mathbf{x}_v}{\|\mathbf{x}_v\|} \right\rangle,$$

we divide all the nodes into K groups of functional clusters by employing the K-medoids algorithm [29] due to its robustness. Formally, we maximize the following objective function with respect to a set of medoids (representative nodes) $R \subset V$:

$$f(R) = \sum_{v \in V} \max_{r \in R} \rho(v, r). \tag{3}$$

To maximize objective function $f(R)$, we employ a greedy algorithm using the following marginal gain with respect to each candidate node w by fixing set R of the already selected medoids:

$$\begin{aligned} g(w; R) &= f(R \cup \{w\}) - f(R) \\ &= \sum_{v \in V \setminus R} \max\{\rho(v, w) - \mu(v; R), 0\}, \end{aligned} \tag{4}$$

where $\mu(v\ ; R) = \max_{r\in R}\{\rho(v,r)\}$ if $R \neq \emptyset$; otherwise $\mu(v\ ; \emptyset) = 0$. Then we can summarize the greedy algorithm. After initializing $k \leftarrow 1$ and $R_0 \leftarrow \emptyset$, we repeatedly select and add each medoid by

$$\hat{r}_k = \underset{w\in V\setminus R_{k-1}}{\arg\max}\ g(w; R_{k-1}),\ \ R_k \leftarrow R_{k-1} \cup \{\hat{r}_k\}$$

during $k \leq K$ with increment $k \leftarrow k+1$. From the obtained K medoids $R = \{r_1, \cdots, r_K\}$, we calculate each functional cluster:

$$V^{(k)} = \{v \in V; r_k = \underset{r\in R}{\arg\max}\{\rho(v,r)\}\}.$$

Due to the submodularity of the objective function, we are guaranteed to obtain a unique greedy solution with reasonably high quality [30], unlike such other standard methods as K-means clustering. Moreover, in the case of our problem setting, by setting $\mathbf{x}_v \leftarrow \mathbf{x}_v/\|\mathbf{x}_v\|$ for each node $v \in V$, we derive the following transformations:

$$g(w; \emptyset) = f(\{w\}) = \sum_{v\in V} \rho(v, w) = \left\langle \mathbf{x}_w, \sum_{v\in V} \mathbf{x}_v \right\rangle. \tag{5}$$

Thus, we can efficiently obtain the first medoid, $\hat{r}_1 = \arg\max_{w\in V} g(w; \emptyset)$, with computational cost $O(NS)$. In our approach, we employ an arbitrary similarity definition without restricting the cosine similarity. One computational advantage of using the cosine similarity is that we can efficiently obtain $\hat{r}_1$, as described above.

4 Pivot-Based Acceleration

When N functional vectors are large and the space on the main memory is insufficient, we need to recalculate most of the $N(N-1)/2$ distances (similarities) for all K (> 2) greedy steps at the K-medoids clustering phase, which amounts to a computational cost of $O(KN^2S)$. To overcome this problem, we propose a new technique for accelerating the K-medoids clustering phase by combining the lazy evaluation and pivot pruning techniques. As for the lazy evaluation technique, if some condition based on the submodular property is satisfied with respect to a candidate node $w \in \mathcal{V} \setminus \mathcal{R}$, we can skip the actual computation of marginal gain $g(w; \mathcal{R})$ and simultaneously avoid a large number of the corresponding similarity calculations. On the other hand, as for the pivot pruning techniques, if some condition based on the triangle inequality is satisfied with respect to a pair of nodes $w, x \in \mathcal{V} \setminus \mathcal{R}$, we can skip the actual computation of the corresponding similarity calculation $\rho(w, x)$. As representative ones, we employ the outlier and medoid pivots selection techniques where the former pivots are selected as some objects having larger distances to other objects, while the latter pivots as those having smaller ones.

In the lazy evaluation technique [31], which is applied at the k-th medoid selection step, we utilize an upper bound value $UB(w)$ of marginal gain $g(w; R)$ for each candidate node $w \in V$. More specifically, after initializing $UB(w) \leftarrow g(w; \emptyset)$, which is calculated in Eq. (5), we update $UB(w) \leftarrow g(w; R_h)$ when $g(w; R_h)$ is actually calculated at the h-th medoid selection step. Evidently, due to the submodular property, it is guaranteed that $g(w; R_k) \leq UB(w)$ for $k > h$. Let g_k^* be the current best marginal gain at the selection step for obtaining the k-th medoid; then we can omit the calculation of $g(w; R_k)$ when $UB(w) \leq g_k^*$. On the other hand, to obtain better g_k^* at an earlier stage, we sort the candidate nodes in descending order with respect to $UB(w)$ and evaluate them from the top of the sorted list.

In the pivot pruning technique [32], which is applied by actually calculating $g(w; R_k)$, we utilize lower bound distance $LB(w, v; P)$ of distance $d(w, v)$ to examine pruning condition $\rho(w, v) \leq \mu(v; R)$, where $P \subset V$ is a set of pivots described below and $d(w, v)$ is a standard Euclidean distance obtained as $d(w, v) = \sqrt{1 - \rho(w, v)}$. From Eq. (4), we do not add any value when pruning condition $\rho(w, v) \leq \mu(v; R)$ holds. More specifically, from the triangle inequality, we can utilize the following lower bound distance $LB(w, v; P)$:

$$LB(w, v; P) = \max_{p \in P} |d(w, p) - d(v, p)| \leq d(w, v).$$

Thus, when $\sqrt{1 - \mu(v; R)} \leq LB(w, v; P)$ and noting that

$$\sqrt{1 - \mu(v; R)} \leq LB(w, v; P) \leq d(w, v) = \sqrt{1 - \rho(w, v)},$$

pruning condition $\rho(w, v) \leq \mu(v; R)$ holds without actually calculating $\rho(w, v)$. Now, we introduce two types of pivots P so that the pivot pruning technique works adequately. As the first type of pivot, we utilize the obtained medoids as pivots; after setting $P \leftarrow \{r_1\}$, we successively add obtained medoid r_k as a pivot by $P \leftarrow P \cup \{r_k\}$. In the second type of pivot, we select outlier objects as pivots. With the first medoid r_1, we select and add the first outlier pivot by

$$\hat{q}_1 = \arg\max_{v \in V} d(v, r_1), \;\; P \leftarrow P \cup \{\hat{q}_1\}.$$

Then we select and add the h-th pivot by

$$\hat{q}_h = \arg\max_{v \in V} \min_{p \in P} d(v, p), \;\; P \leftarrow P \cup \{\hat{q}_h\}.$$

We denote the maximum number of outlier pivots by H, and in our proposed algorithm, calculate them before selecting the second medoid, r_2.

Hereafter, the lazy evaluation technique, the pruning technique by medoids, and the pruning technique by the outlier pivots are called the LE, MP, and OP techniques, respectively. In our proposed method, we apply the LE technique prior to the pivot pruning techniques. This is because when the marginal gain calculation of $g(w; R)$ is omitted by the LE technique, we can simultaneously prune all of the similarity calculations of $\rho(w, v)$ for any $v \in V$. On the other hand, in our implementation, we apply the MP technique prior to the OP technique, because as shown below in our experiments, at the k-medoid selection step, a combination of the LE and MP techniques achieved a reasonably high performance when the iterative step of clustering proceeds. We summarize the entire flow of our proposed algorithm as follows:

1. Select the first medoid, r_1;
2. Select outlier pivots $P = \{p_1, \ldots, p_H\}$;
3. Repeat the following steps from $k = 2$ to K with k's increments:
    (a) Examine the pruning conditions, LE, MP, and OP, in this order;
    (b) Calculate the similarities and marginal gains for the unpruned nodes and extract the k-th medoid;

5 Experimental Design

5.1 Datasets

We used OpenStreetMap (OSM) data of the following 12 cities from Metro Extracts[1] in August, 2015: Shizuoka prefecture, Shizuoka city, Kanagawa, Kyoto, San Francisco, New York, Barcelona, Seoul, Brasilia, Washington D.C., Cairo, and New Delhi. Hereafter, we abbreviate Shizuoka prefecture and Shizuoka city as ShizuokaP and ShizuokaC, respectively. Some of these cities were selected as a subset of those previously studied [2], but note that in our experiments, each area of these cities covers a wide area around the city that is more than 100 times larger than the 1-square mile area used in the previous study [2]. Then we extracted all of the highways and all the nodes from the OSM data of each city and constructed each spatial network by mapping the ends and the intersections of the streets into nodes and the streets between nodes into links. To simplify our analyses, we deleted the nodes used for representing the curved segments of highways by directly connecting both sides of the deleted ones.

Table 2 shows the basic statistics of the networks for the 12 cities, where C and L respectively denote the averages of the clustering coefficient and the shortest path length over each network. Although the area and the numbers of nodes and links $|V|$ and $|E|$ are substantially different, the degree distributions defined by p_j as well as C and L are quite similar as common characteristics of these spatial networks.

[1] https://mapzen.com/data/metro-extracts.

Table 2 Basic statistics as network

City	Area	$\|V\|$	$\|E\|$	p_1	p_2	p_3	p_4	$p_{>4}$	C	L
ShizuokaP	155 ×119 km	110,925	162,322	0.121	0.070	0.576	0.228	0.005	0.05	83.09
ShizuokaC	50 ×87 km	31,041	43,608	0.186	0.032	0.572	0.204	0.005	0.04	48.68
Kanagawa	80 ×60 km	295,151	402,576	0.192	0.051	0.597	0.155	0.005	0.04	129.17
Kyoto	110 ×119 km	88,800	128,601	0.099	0.090	0.633	0.174	0.004	0.07	103.43
San Francisco	90 ×50 km	110,700	156,821	0.173	0.037	0.583	0.199	0.009	0.05	79.43
New York	130 ×95 km	325,962	466,510	0.159	0.033	0.600	0.204	0.004	0.04	103.14
Barcelona	45 ×30 km	66,790	99,387	0.103	0.031	0.659	0.201	0.006	0.06	53.07
Seoul	100 ×123 km	103,444	150,822	0.111	0.078	0.605	0.198	0.008	0.04	43.99
Brasilia	120 ×104 km	95,811	136,955	0.133	0.025	0.694	0.146	0.002	0.04	92.94
Washington D.C.	23 ×18 km	24,564	38,053	0.096	0.028	0.571	0.293	0.012	0.05	51.89
Cairo	87 ×86 km	56,781	85,594	0.068	0.025	0.733	0.172	0.002	0.04	58.80
New Delhi	109 ×75 km	116,905	166,743	0.138	0.017	0.702	0.142	0.002	0.03	78.04

5.2 Baseline Methods

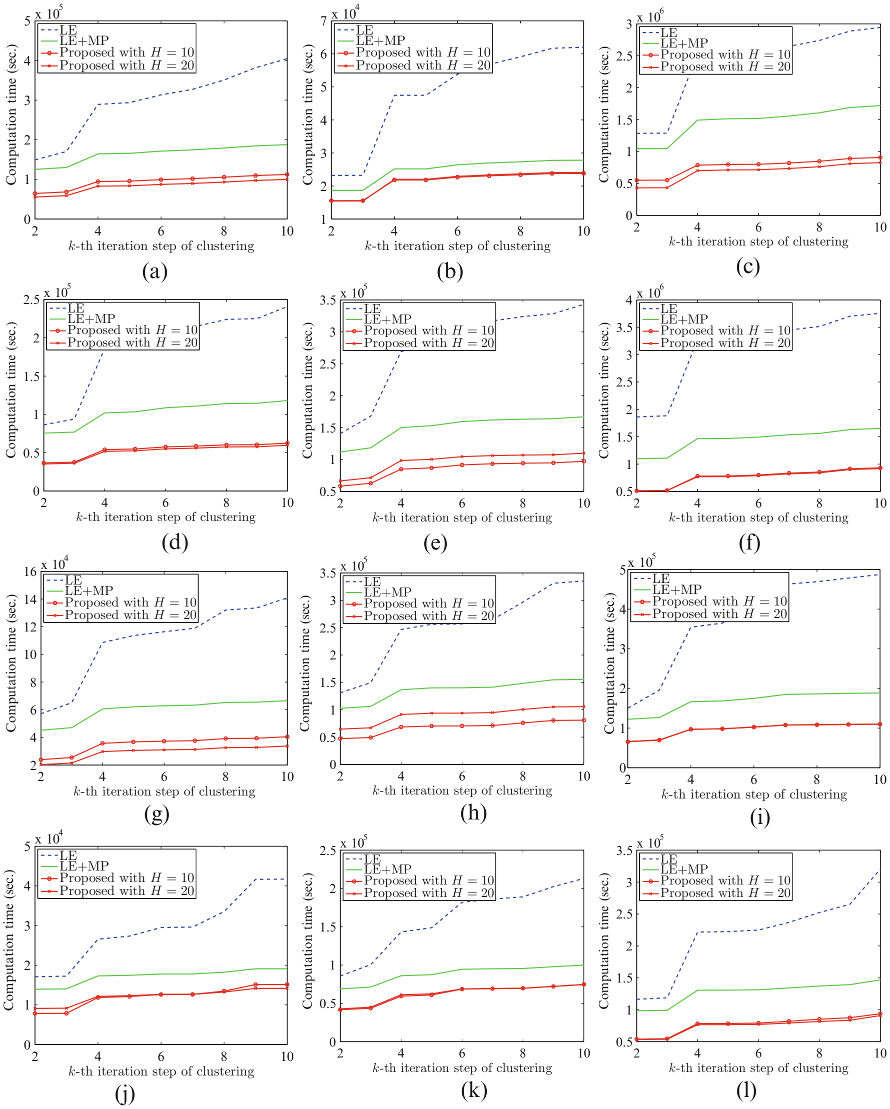
In our experiments, we evaluated the computational efficiency of our proposed method under the setting of the dimensionality of functional vector $S = 10{,}000$ and number of clusters $K = 5,\ 10$ and compared it to the following two methods: the first method only employed the LE technique, the (a) LE method, and the second method employed both the LE and MP techniques, the (b) LE+MP method. Our method employs all the LE, MP, and OP techniques, which is simply referred to as (c) Proposed method. In the Proposed method, we changed the number H of the outlier pivots to 10 or 20. We performed our experiments on a computer system with a Xeon processor E5-2697 2.7 GHz and 256-GB main memory.

6 Evaluation of Computational Performance

Figure 2 compares the computation times of the LE, LE + MP, and Proposed methods with respect to the networks of the 12 cities, where the horizontal and vertical axes, respectively, stand for the number of medoids (clusters) and the computation times. In this figure, we only show the computation times of the k-medoids clustering phase and change the number of outlier pivots as $H = 10, 20$ in the Proposed method. First, from Fig. 2, for all the networks, we confirmed that the Proposed method worked substantially faster than the other LE and LE + MP methods. The Proposed method achieved from three to seven times better performance than the LE method, which is a state-of-the-art method. Especially for $k = 4$, the Proposed method needs far less computation times than the LE method, and as for $k > 4$, the computation times of the Proposed method increase very little while those of the LE method substantially increase.

Second, the Proposed method also worked faster than the LE + MP method. Generally speaking, medoids exist in the center of each cluster in a feature space (functional vector space). On the other hand, in the Proposed method, we selected outlier functional vectors as pivots that are located far from the center of gravity or the already selected nearest pivot. Since outliers effectively reduce the number of similarity calculations, we can obtain these results. Third, our experimental results indicate that the desirable number of outlier pivots in the Proposed method depends on the dataset.

Next we evaluated the effects of the three pruning techniques in the Proposed method that accelerated the clustering phase. For each k-th medoid selection step, let $LE(k)$, $MP(k)$, and $OP(k)$ be the sets of node pairs whose actual similarity calculations are omitted by the LE, MP, and OP techniques. Recall that in the Proposed method, the LE, MP, and OP techniques are applied in this order. Thus, their actual pruning rates are calculated as $\alpha(LE(k)) = |LE(k)|/N^2$, $\alpha(MP(k)) = (|LE(k) \cup MP(k)| - |LE(k)|)/N^2$, and $\alpha(OP(k)) = (|LE(k) \cup MP(k) \cup OP(k)| - |LE(k) \cup MP(k)|)/N^2$, respectively.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Fig. 2 Computation times for 12 cities, where blue dashed, green solid, and red solid lines with circles or crosses, respectively, stand for LE, LE+MP, and Proposed methods. (**a**) ShizuokaP. (**b**) ShizuokaC. (**c**) Kanagawa. (**d**) Kyoto. (**e**) San Francisco. (**f**) New York. (**g**) Barcelona. (**h**) Seoul. (**i**) Brasilia. (**j**) Washington D.C. (**k**) Cairo. (**l**) New Delhi

Figure 3 compares these pruning rates of the k-th medoid selection step by changing $k = 2$–10, where the blue, green, and red bars, respectively, stand for the pruning rates of $\alpha(LE(k))$, $\alpha(MP(k))$, and $\alpha(OP(k))$, and we show the results of $\alpha(OP(k))$ at $H = 20$. Recall that our method calculates the first medoid, r_1, by Eq. (5). From Fig. 3, for all the networks, the LE technique did not omit any marginal

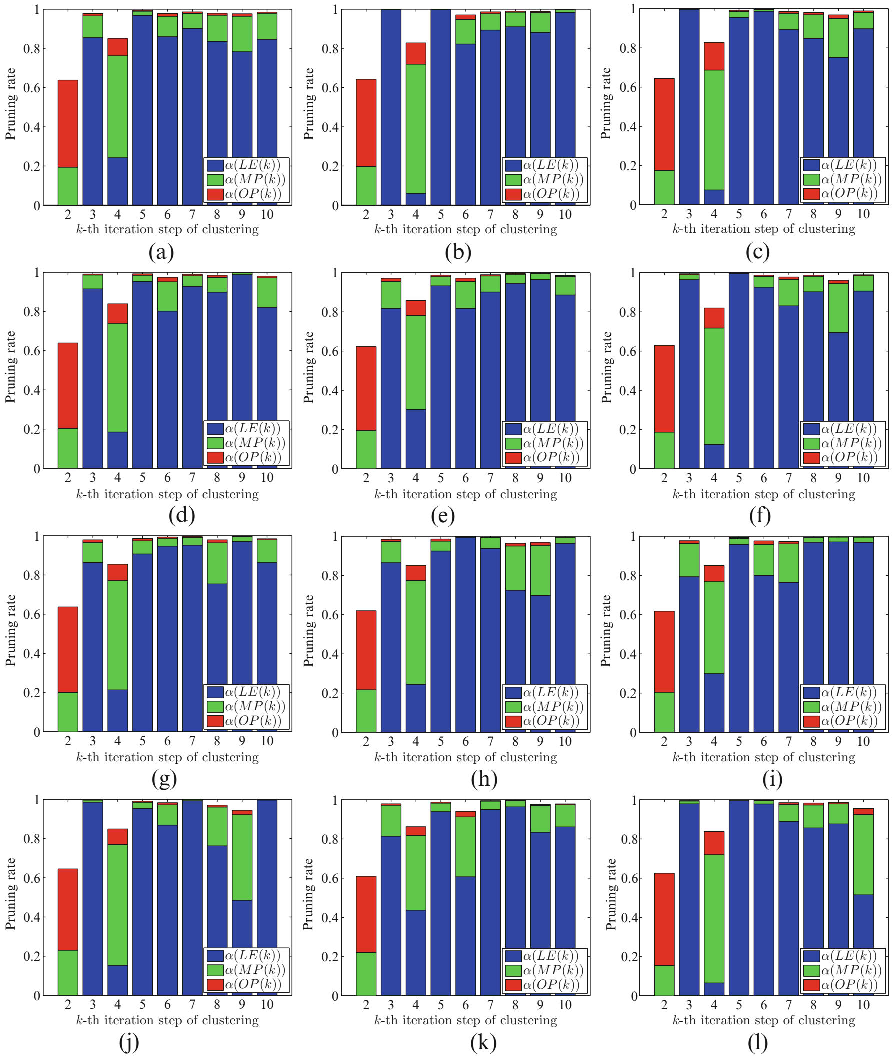


Fig. 3 Pruning rates for 12 cities, where blue, green, and red bars stand for $\alpha(LE(k))$, $\alpha(MP(k))$, and $\alpha(OP(k))$, respectively. (**a**) ShizuokaP. (**b**) ShizuokaC. (**c**) Kanagawa. (**d**) Kyoto. (**e**) San Francisco. (**f**) New York. (**g**) Barcelona. (**h**) Seoul. (**i**) Brasilia. (**j**) Washington D.C. (**k**) Cairo. (**l**) New Delhi

gain calculation at step $k = 2$ and also worked quite poorly at step $k = 4$. This result indicates that each upper bound $UB(w)$ was a quite rough approximation to actual marginal gain $g(w; R)$ at these steps. The MP technique also showed relatively poor pruning rates at step $k = 2$ because the MP technique just used one pivot. Therefore, by applying the OP technique, the Proposed method stably achieved reasonably high pruning rates.

7 Evaluation of Extracted Functional Clusters

Figure 4 shows the visualization results of the functional clusters at $K = 5$ where our experimental results are demonstrated only by using $K = 5$, because we could obtain consistent results at the other numbers of clusters. In these figures, the functional clusters $V^{(1)}, \cdots, V^{(5)}$ are depicted by using different colors of red, green, blue, yellow, and magenta in this order. As remarkable characteristics of these results, we can see that all the cities share the following similar characteristics: the blue regions ($V^{(3)}$) are typically surrounded by the red regions ($V^{(1)}$), and the other regions of green ($V^{(2)}$), yellow ($V^{(4)}$) or magenta ($V^{(5)}$) likely surround the red regions.

In case of the ShizuokaP results shown in Fig. 4a, which is one of familiar cities for authors, the blue, red, magenta, green, and yellow regions are distributed from the center of the main cities to mountainous areas in this order. Each blue region approximately corresponds to the central area of each city, where at least one railway station exists. Also, the red regions mainly contain nodes whose degree is three and exist around each blue region. Based on these observations, we refer to the blue and red regions as the downtown and suburban areas. Similarly, the green regions contain many nodes, whose degree is one, that exist in agricultural areas or at the foothills of mountains. The yellow regions contain many nodes whose degree is two, which mean long continuous roads to other towns over mountainous areas. A similar tendency can be seen in other cities used in our experiments: Fig. 4b–l. These observations, which are naturally interpretable from the aspects of geographical restrictions, suggest the practical usefulness of our method. As another advantage of our visualization results, we can intuitively understand the detailed regions of each city in terms of the characteristics of interpretable functional clusters.

Figure 4b–l suggest quite similar explanations for the results of the other eleven cities, as discussed for ShizuokaP, especially for the first three functional clusters: ($V^{(1)}$: red, $V^{(2)}$: green, and $V^{(3)}$: blue regions). Note that due to the property of the greedy algorithm used in the Proposed method, which computes a new medoid by fixing previously selected medoids, a new functional cluster is usually formed by splitting and specifying the existing clusters. For example, we can trace the result that the 5th functional clusters ($V^{(5)}$: magenta regions) of San Francisco (Fig. 4e) and Washington D.C. (Fig. 4j) are formed from the second one ($V^{(2)}$: green regions) and the first one ($V^{(1)}$: red regions), respectively. The former seems to be mountainous areas characterized by cul-de-sacs containing many nodes with degree 1, while the latter corresponds to suburban areas characterized by three-way junctions containing many nodes with degree 3; therefore, we consider that these 5th functional clusters can be parts of individual characteristics of these cities.

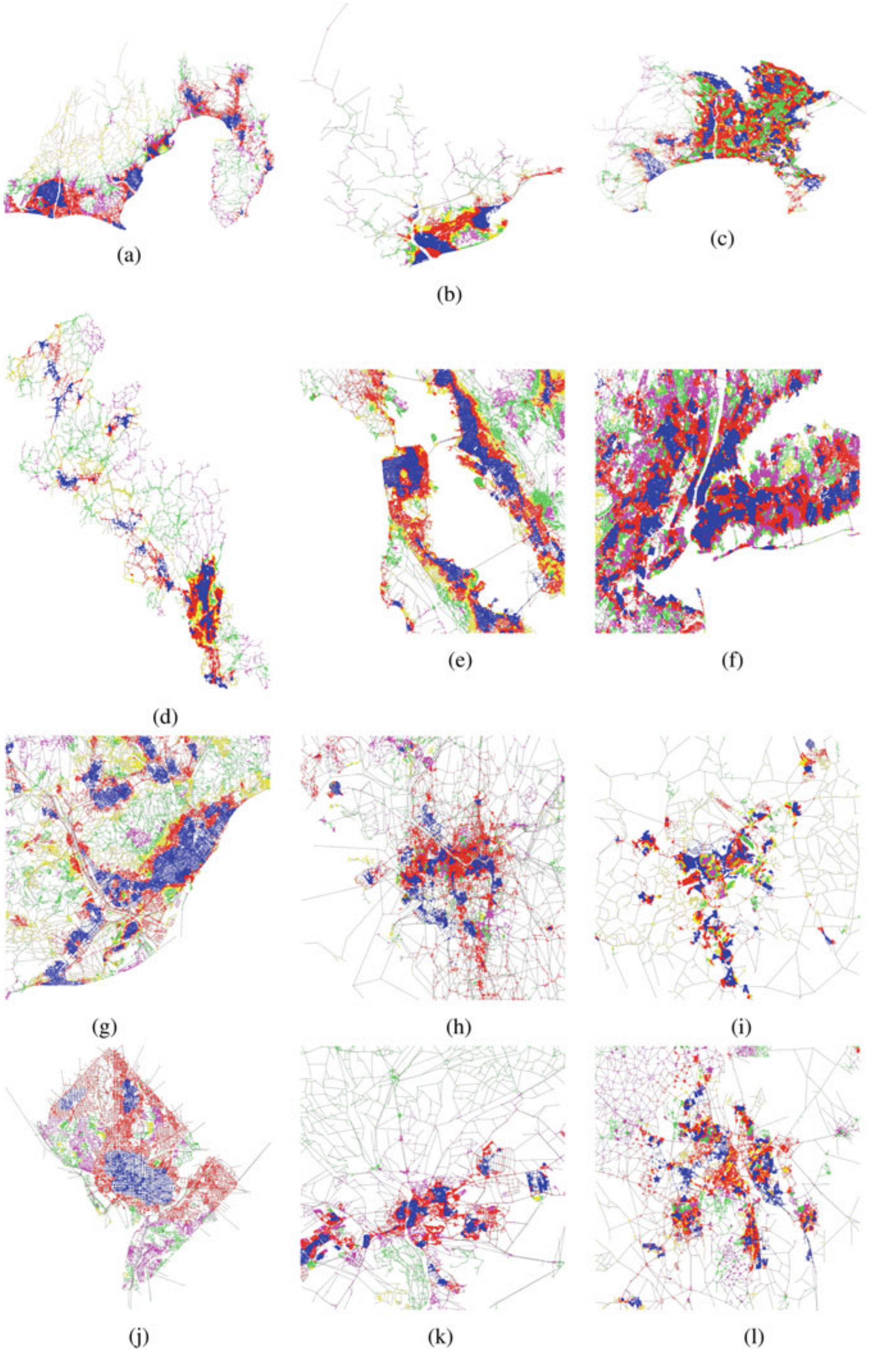


Fig. 4 Visualization results of functional clusters. (**a**) ShizuokaP. (**b**) ShizuokaC. (**c**) Kanagawa. (**d**) Kyoto. (**e**) San Francisco. (**f**) New York. (**g**) Barcelona. (**h**) Seoul. (**i**) Brasilia. (**j**) Washington D.C. (**k**) Cairo. (**l**) New Delhi