Simultaneous Detection of Multiple Areas-of-Interest Using Geospatial Data from an Online Food Delivery Platform (Industrial Paper)

Bingnan Li[1], Liying Chen[2], Daping Xiong[2], Shuiping Chen[2], Renqing He[2], Zhizhao Sun[2], Samsung Lim[1], Hai Jiang[3,*]

[1]School of Civil and Environmental Engineering, University of New South Wales, Sydney, NSW 2052, Australia

[2]Meituan, No. 4 Wangjing East Street, Chaoyang District, Beijing 100020, China

[3]Department of Industrial Engineering, Tsinghua University, Beijing 100084, China

haijiang@tsinghua.edu.cn

ABSTRACT

With the development of mobile Internet, online food delivery (OFD) services have become increasingly popular in our daily lives. OFD platforms rely heavily on accurate Areas-of-Interest (AOIs) information on many aspects of their operations to pinpoint customers' exact locations and to define the service areas of restaurants. Recently, OFD platforms have started to tap into the vast amount of geospatial data generated in their day-to-day business to improve the accuracy of their AOI information. Although there has been a proliferation of studies that leverage such data to detect the underlying AOIs, for example, to identify the names and spatial boundaries of the AOIs, they focus on the single-AOI detection problem, that is, they detect AOIs one at a time and ignore their spatial dependency. This would end up with inconsistent results, i.e., AOIs with overlapping spatial boundaries. To address this issue, we propose a new approach to detect multiple AOIs simultaneously and solve the multi-AOIs detection problem. In our approach, we first apply the existing single-AOI detection algorithms to generate candidate spatial boundaries for AOIs in a neighborhood, and then develop a Binary Integer Linear Programming (BILP) model to determine the best candidate spatial boundaries for these AOIs while accounting for their spatial dependency. We conduct numerical experiments using real data from Meituan, the largest OFD platform in China. Results show that our model not only produces consistent AOI boundaries, but also improves the average F_1 score by 4.7%.

CCS CONCEPTS

• **Information systems → Data mining**.

KEYWORDS

online food delivery (OFD), area-of-interest (AOI), multi-AOIs detection, optimization model

*Corresponding author.

SIGSPATIAL '22, November 1–4, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9529-8/22/11.
https://doi.org/10.1145/3557915.3561014

ACM Reference Format:
Bingnan Li[1], Liying Chen[2], Daping Xiong[2], Shuiping Chen[2], Renqing He[2], Zhizhao Sun[2], Samsung Lim[1], Hai Jiang[3,*]. 2022. Simultaneous Detection of Multiple Areas-of-Interest Using Geospatial Data from an Online Food Delivery Platform (Industrial Paper). In *The 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22), November 1–4, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3557915.3561014

1 INTRODUCTION

With the development of mobile Internet, online food delivery (OFD) services have become more and more popular in our daily lives [39]. Globally, the OFD market's revenue had increased by 27% year-on-year and reached $136.4 billion in 2020. The number of users had increased by nearly 25%, reaching 1.46 billion in 2020 [20]. China is leading the way in terms of market size and has achieved great success. Take Meituan, one of the world's largest OFD platforms as an example [23]. The total revenue had increased by nearly 18% year-on-year and reached 114.8 billion yuan in 2020. The number of users had increased by 13%, reaching 460 million in 2019 [27].

OFD platforms rely heavily on accurate Areas-of-Interest (AOIs) information in their operations. An AOI, also known as a region-of-interest (ROI), refers to a polygon selection in a map that someone may find useful or interesting, for example, a residential complex, a public park or a shopping mall [38]. OFD platforms are concerned with two key properties associated with an AOI, that is, its name and spatial boundary. Spatial boundaries of AOIs are stored as vector formats, which can be easily used in geospatial analysis and improve service efficiency of the OFD industry. In Figure 1, we give an example of an AOI, whose name is *Yishashi Garden Apartments*, and the spatial boundary is colored in blue. It is a gated apartment complex, and all 10 buildings of it are within the spatial boundary.

An OFD platform uses AOI information in many different ways, for example: (1) When a customer places an order, the OFD platform relies on AOI information to resolve the delivery address. Based on the GPS coordinates of customers and the boundaries of nearby AOIs, the platform would suggest possible AOI names to assist customers to pinpoint their exact locations, which is critical to ensure timely delivery. (2) On an OFD platform, a restaurant, say a McDonald's, is often asked to specify its service area as a list of AOIs. The OFD platform then uses the GPS location of the customer together with the spatial boundaries of the AOIs to determine whether to

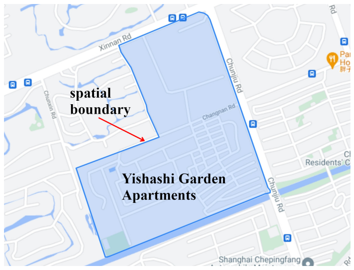


Figure 1: An example of an AOI.

show this McDonald's to the customer. If the OFD platform does not have the accurate spatial boundary for an AOI, for example, if the spatial boundary of *Yishashi Garden Apartments* is incorrect and Building #10 is erroneously excluded, the OFD platform would not allow residents living in Building #10 to order from this McDonald's, although residents living in other buildings of this complex can. This would create inconsistent experience among residents living in the same gated complex.

OFD platforms have spent considerable efforts to improve the accuracy of their AOIs. Recently, they started to tap into the vast amount of geospatial data generated during the ordering process which is illustrated in Figure 2. When a customer takes out the phone to order, the OFD platform typically performs the following procedures.

- In Step 1, the OFD platform gets the GPS coordinates of the customer's location from the GPS chip of the phone, and by using the boundaries of known AOIs, it suggests possible AOI names as delivery addresses.
- In Step 2, if the customer does not modify the suggested AOI name, the AOI is proved to be accurate; otherwise, the modified address and GPS coordinates are sent to the geospatial database for potential AOIs.
- In Step 3, if the customer is not located in a known AOI, he/she is asked to type the address manually, which together with the GPS coordinates become the geospatial data that can be used to detect AOIs for potential AOIs.

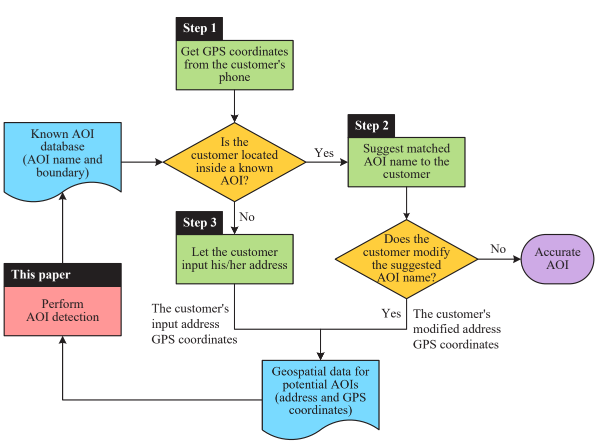


Figure 2: How AOI information is used in ordering process.

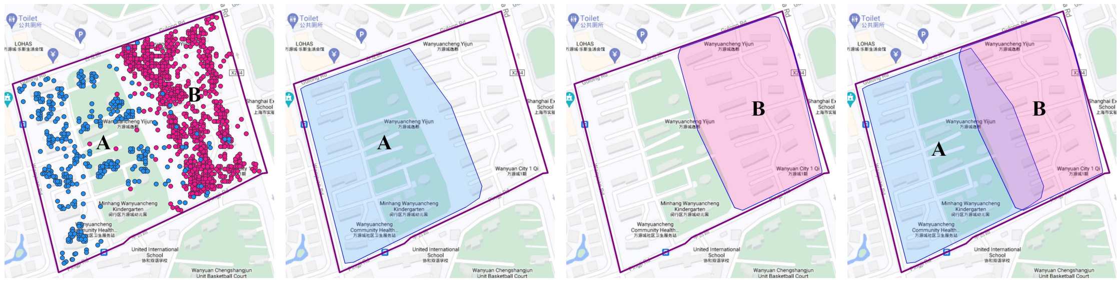
In this paper, we investigate how to detect AOIs from the geospatial data collected in the above process (see the red rectangle in Figure 2). The AOI detection problem involves identifying the name and the boundary of an underlying AOI. Although there has been a proliferation of studies that investigate the AOI detection problem, existing approaches all focus on the single-AOI detection problem, that is, they detect AOIs one at a time. In fact, some noisy GPS points locate in the wrong locations, therefore, they may construct spatial boundaries of different AOIs with overlaps. Among existing studies, multiple geospatial data sources are applied into the single-AOI detection problem, including social media data (e.g., geo-tagged Flicker photos [16, 32, 35] and geo-tagged tweets [4, 9, 26]), remote sensing data [9], and delivery data [33].

Since single AOI detection algorithms detect AOIs independently of each other, they tend to produce inconsistent results. Take Figure 3 as an example. In Figure 3(a), there are two AOIs whose names are *A* and *B*. The blue dots are GPS locations located in AOI *A*, while the red dots are those located in AOI *B*. These GPS locations are generated directly from customers' mobile devices and collected from the OFD platform. Existing studies would first use the blue dots to identify the spatial boundary of AOI *A*, which is shown in Figure 3(b); and then use the red dots to identify the spatial boundary of AOI *B*, which is shown in Figure 3(c). When we overlay the spatial boundaries of these two AOIs in Figure 3(d), they overlap with each other, which is inconsistent. In summary, different approaches of single-AOI detection tend to generate inconsistent results and cannot fully leverage GPS data in adjacent AOIs.

In this research, we aim to address the challenge faced by single-AOI detection models through simultaneously detecting multiple AOIs. We propose to detect multiple AOIs simultaneously, that is, to solve the multi-AOIs detection problem. In our approach, we first apply existing single-AOI detection algorithms to generate candidate spatial boundaries for AOIs in a neighborhood, we then develop a Binary Integer Linear Programming (BILP) model to determine the best candidate spatial boundaries for these AOIs while accounting for their spatial dependency. We conduct numerical experiments using real data from Meituan, the largest OFD platform in China. Results show that our model not only produces consistent AOI boundaries but also improves the average F_1 score by 4.7%. We improve the accuracy and preserve the details of the boundaries of detected AOIs by applying a Hidden Markov Model (HMM) to the road network dataset.

The contributions of this paper can be summarized as follows: (1) To the best of the authors' knowledge, we are the first to investigate the multi-AOIs detection problem. (2) By accounting for the spatial dependency among neighbouring AOIs, we ensure that our approach can produce AOI boundaries that are consistent with each other. (3) We formulate the problem as a Binary Integer Linear Programming (BILP) model, which can be efficiently solved by standard branch-and-bound procedures. (4) Using the optimization model in the dataset collected from Meituan platform, results show that our model identifies Multi-AOIs and improves the average F_1 score from 0.847 to 0.894 and achieves the best average F_1 score among all single-AOI detection methods.

The rest of this paper is organized as follows. In Section 2, the related work of AOI detection is discussed. Section 3 describes in detail of the optimization model. In Section 4, we perform numerical



(a) (b) (c) (d)

Figure 3: Existing studies detect AOIs independently of each other and may produce AOIs with overlaps. (a) shows the GPS locations for customers located in AOIs A and B; (b) and (c) illustrate the estimated spatial boundaries of AOIs A and B, respectively; (d) shows the spatial boundaries of AOIs A and B overlap, which is not acceptable.

experiments to evaluate the benefits of our optimization model. Finally, we make a conclusion and describe the possible future expectations in Section 5.

2 EXISTING STUDIES ON AOI DETECTION

With the development of mobile devices, volunteered geographic information (VGI) is another source of data for numerous spatial applications [15, 34]. In addition, OFD information is a new type of geographic data generated by millions of customers and riders. Although the absence of quality control, some data have been demonstrated the same quality of the authoritative data [24]. Most of the existing studies are based on VGI data, such as geo-tagged Flicker photos, tweets and other social media data [6, 21, 36].

Existing approaches for single-AOI detection can be broadly divided into three groups: pre-defined shapes, density-based clustering, and grid-based aggregation. And some other novel approaches are also introduced in this field.

Pre-defined shapes. In [32], the authors use circle with fixed radius to extract popular tourist routes based on geo-tagged Flickr photos. In particular, circles are used to represent a trajectory of coordinates into a series of AOIs. In [1], the authors use rectangular AOIs to represent stadiums in a study of trajectory pattern mining. Specifically, a stadium's AOI is the minimum rectangle of its area. Due to the lack of ability to construct complex polygons, this approach has significant limitations in practical application.

Density-based clustering. In [19], the authors use K-Means to discover landmarks based on geo-tagged photos. This method needs to set the number of clusters, but it is difficult to determine an optimal number beforehand. In [8], the authors apply the algorithm of Mean Shift [7] to cluster the locations based on a group of geo-tagged Flicker photos. Rather than setting the number of clusters, this algorithm needs to specify a value to determine the density radius, which is difficult to accurately discover proper number for different areas. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a widely used algorithm of density-based clustering for geospatial data [13]. In [16], the authors apply DBSCAN clustering algorithm to identify urban AOIs based on Flickr geo-tagged photos. In [33], the authors make use of DBSCAN algorithm to identify clusters based on delivery addresses. Compared to above clustering algorithms, DBSCAN is very robust against outliers, and there are no constraints on the boundaries of clusters. Furthermore, a number of studies have proposed improvements of clustering approaches based on DBSCAN. In [36], the authors apply the algorithm of P-DBSCAN to eliminate noises of geographic coordinates and investigate the tourists' behaviors in Hong Kong. In [30], the authors introduce the algorithm of C-DBSCAN, which defines the constraints based on the background knowledge. In [26], the authors propose M-DBSCAN to reduce the uncertainty of detecting clusters by DBSCAN based on different density and cluster size scales. H-DBSCAN [5] is introduced by improving and integrating DBSCAN and OPTICS [2]. In [18], the authors apply H-DBSCAN algorithm to identify AOIs and interest patterns of tourists from Flickr geo-tagged photos in Vienna. Other clustering methods are also applied to discover clusters of AOIs. In [25], the authors devise a clustering method for discovering AOIs from image densities and enhanced by the secondary densities of sites adjacent to the images. In [35], the authors propose an adaptive urban clustering method to discover Points-of-Interest (POIs) based on different granularities.

Grid-based aggregation. In [22], the authors map coordinates into a grid cell and defined temporal constraints to discover AOIs. In [31], the authors collect geo-tagged photos with location names and conduct clustering by Delaunay triangulation, then POI was recognized as the average coordinates located in the cluster. In [28], the authors propose a grid-based algorithm to solve the problem of discovering Geometries-of-Interest (GOIs) of moving objects based on GPS trajectories.

Besides above three groups of approaches, we also have reviewed other single-AOI detection algorithms. In [10], the authors introduce a grid-based Integer Linear Programming (ILP) model to discover AOIs. In [4], the authors propose an algorithm of G-ROI for discovering ROIs on multiple social media datasets. The G-ROI contains two steps of reduction and selection, and achieves higher F_1 score compared with other methods.

The boundary of a cluster is often constructed by a convex hull [3]. Given a set of points on a Cartesian plane, the convex hull of these points is represented by the external polygon on them [21]. In order to match boundaries better and reduce blank parts for polygon construction, the alpha-shape [12] and other approaches of building concave hull are widely used [28]. The algorithm of alpha-shape is built on the base of Delaunay triangulation and can be used to construct polygons with different shapes flexibly. In [11], the authors propose an algorithm of chi-shape to construct a concave hull, and [16] use this method to construct AOIs based on geo-tagged Flicker photos.

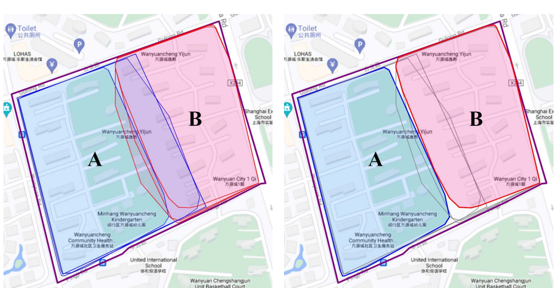
In summary, our study is different from earlier research in two ways. First, our approach fully utilizes mutual exclusion of different AOIs and combines multiple single-AOI detection algorithms together; Second, instead of focusing on detecting one particular AOI, we propose an optimization model which can detects multi-AOIs simultaneously.

3 METHODOLOGY

We develop an optimization model for the simultaneous detection of multi-AOIs in this section. Inputs to this model include GPS points and multiple candidate spatial boundaries constructed by single-AOI detection models. The optimization model then outputs the optimal spatial boundaries of AOIs. We first use an example to illustrate how the model works, and then detail the optimization model. Finally, we refine the model by introducing the geohash technique.

3.1 The General Idea

In order for readers to understand our approach better, before diving into the details, we describe how the model works. We continue using the example mentioned in Section 1 to explain the working strategy of the optimization model. Figure 3(d) shows candidate spatial boundaries of AOIs A and B by a single-AOI detection algorithm. If we change the algorithms and their parameter values, different candidate spatial boundaries can be generated. For AOIs A and B, two sets of candidate sptial boundaries are generated as $\Psi_A = \{A_1, A_2, \cdots, A_n\}$ and $\Psi_B = \{B_1, B_2, \cdots, B_n\}$ by n single-AOI detection models. Then these data are fed into the optimization model.



(a) Candidate spatial boundaries (b) Optimal spatial boundaries

Figure 4: Visualization of the proposed approach.

Based on the mutual exclusion of different AOIs, the optimization model helps us discover the optimal spatial boundaries of AOIs A and B from Ψ_A and Ψ_B simultaneously. In this study, the definition of optimal spatial boundaries of AOIs is that spatial boundaries contain the corresponding GPS points as many as possible and there are no overlaps between any two of these spatial boundaries. Figure 4 illustrates the process of our proposed approach. In Figure 4(a), the blue and red lines represent candidate spatial boundaries of Ψ_A and Ψ_B, respectively. In Figure 4(b), the optimal spatial boundaries of AOIs A and B are represented as the thick blue and red boundaries, and other candidate spatial boundaries are colored in gray. The optimal spatial boundaries have no overlaps between each other and represent major regions of AOIs A and B.

Then the framework of the proposed approach is illustrated in Figure 5. First, geospatial data as inputs are fed into different single-AOI detection models, which are based on algorithms of G-ROI (detailed in Section 4.3.1), and DBSCAN (detailed in Section 4.3.2). Then different parameter values are assigned to single-AOI detection models to construct n models, which are labelled as Model-1, Model-2, $\cdots$, Model-n. Then these models are used to construct n candidate spatial boundaries of every AOI. Suppose there are m AOIs in the research region, and they are represented as AOI A, AOI B, $\cdots$, AOI M. From Figure 5, we can see that $m \times n$ candidate spatial boundaries are constructed by different models. To make it easier to understand, we use different fill colors to represent candidate spatial boundaries constructed by different models. For example, candidate spatial boundaries created by Model-1 (i.e., $A_1, B_1, \cdots, M_1$) are filled with blue. And candidate spatial boundaries created by Model-2 and Model-n are filled with pink and green, respectively. Then all these candidate spatial boundaries are fed into the multi-AOIs detection model. After the process of the optimization model, finally, a set of optimal spatial boundaries of AOIs are discovered. For optimal spatial boundaries, different colors of AOIs are corresponded to different single AOI detection models.

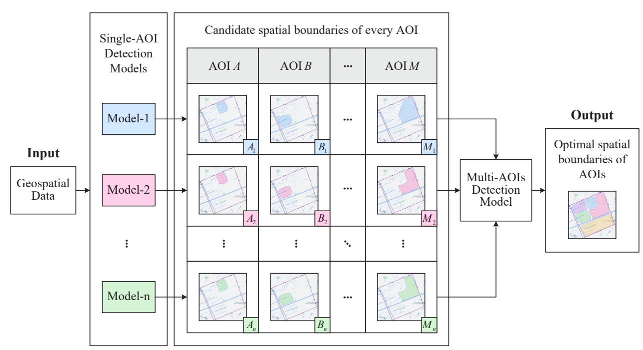


Figure 5: The Framework of the proposed approach.

3.2 Optimization Model

3.2.1 Notation and Terminology. Suppose A_i is one of the AOIs in our research area, where $i \in I$, and $C_{i,j}$ is one of the candidate spatial boundaries of A_i, where $j \in J$. Suppose P_k is one of the GPS points, where $k \in K$. And whether the GPS point P_k is located in $C_{i,j}$ or not is denoted by a binary variable $\delta_{i,j,k}$:

$$\delta_{i,j,k} = \begin{cases} 1, & \text{if the GPS point } P_k \text{ is located in } C_{i,j}; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Then we use a binary variable $\gamma_{k,i}$ to indicate whether the corresponding AOI name of P_k is A_i, if the answer is yes, then $\gamma_{k,i} = 1$, otherwise, $\gamma_{k,i} = 0$. That is,

$$\gamma_{k,i} = \begin{cases} 1, & \text{if } P_k\text{'s corresponding AOI name is } A_i; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

If the candidate spatial boundary $C_{i,j}$ is chosen as the optimal spatial boundary of A_i by the optimization model, then $x_{i,j} = 1$, otherwise, $x_{i,j} = 0$. That is,

$$x_{i,j} = \begin{cases} 1, & \text{if } C_{i,j} \text{ is chosen as the optimal boundary;} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

3.2.2 The Constraints. For any GPS point P_k and all candidate spatial boundaries containing it, at most one candidate spatial boundary can be chosen as the optimal spatial boundary. We apply this requirement by the following constraint:

$$\sum_{i \in I} \sum_{j \in J} \delta_{i,j,k} \cdot x_{i,j} \leq 1, \quad \forall k \in K. \tag{4}$$

For any AOI A_i, only one of the candidate spatial boundaries can be chosen as the optimal spatial boundary. This is achieved through the following constraints:

$$\sum_{j \in J} x_{i,j} = 1, \quad \forall i \in I. \tag{5}$$

$$x_{i,j} \in \{0, 1\}, \quad \forall i \in I, j \in J. \tag{6}$$

3.2.3 Model Formulation. The objective of this study can be regarded as a maximization problem. Thus, this problem is formulated as a BILP model, which is shown as follows:

$$\max \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \delta_{i,j,k} \cdot \gamma_{k,i} \cdot x_{i,j}, \tag{7}$$

subject to constrains (4) through (6).

At last, the optimization model selects the optimal spatial boundaries for AOIs simultaneously.

3.3 Use Geohash to Improve Computational Performance

In Section 3.2, we have to set variables $\delta_{i,j,k}$ and $\gamma_{k,i}$ based on every GPS point P_k, but in fact many GPS points are located in the very close or even the same location. This fact can directly result in many GPS points being computed repeatedly and reducing computational efficiency in the optimization model. Therefore, we can make a grid with equal sized rectangular shaped cells with fine granularity, and every grid cell represents the whole GPS points located in it. For easier implementation of the grid, geohash technique is introduced in our study.

Geohash is a method of encoding geographic points into strings representing cells on the map. The size of a geohash cell is determined by a non-negative integer precision factor. In this research, we choose the precision factor with 9 and it can create square cells with 4.8m × 4.8m. Then all GPS points located in the same geohash cells and with the same parsed AOI names are merged together, and represented as one geohash cell. And we also count the whole orders located in every geohash cell as a weighted value. Finally, we use geohash cells to refine the modelling strategy.

Instead of setting variable $\delta_{i,j,k}$ based on P_k, we introduce G_l as one of the geohash cells, where $l \in L$. The centroid of a geohash cell means the geometric center of the cell. And whether the centroid of the geohash cell G_l is located in $C_{i,j}$ or not is denoted by a binary variable $\delta_{i,j,l}$:

$$\delta_{i,j,l} = \begin{cases} 1, & \text{if the centroid of } G_l \text{ is located in } C_{i,j}; \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

For the geohash cell G_l, we also count the total number of orders located in it as a weighted value w_l. This is denoted by a non-negative integer variable $\beta_{i,j,l}$:

$$\beta_{i,j,l} = \begin{cases} w_l, & \text{if the centorid of } G_l \text{ is located in } C_{i,j}; \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Then we define a binary variable $\gamma_{l,i}$ to determine whether the corresponding AOI name of G_l is A_i, if the answer is yes, then $\gamma_{l,i} = 1$, otherwise, $\gamma_{l,i} = 0$. That is,

$$\gamma_{l,i} = \begin{cases} 1, & \text{if } G_l\text{'s corresponding AOI name is } A_i; \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

For any geohash cell G_l and all candidate spatial boundaries containing it, at most one candidate spatial boundary can be chosen as the optimal spatial boundary. We apply this requirement by the following constraint:

$$\sum_{i \in I} \sum_{j \in J} \delta_{i,j,l} \cdot x_{i,j} \leq 1, \quad \forall l \in L. \tag{11}$$

The definition of $x_{i,j}$ and constraints (5) and (6) remain unchanged. And the objective function is changed as follows:

$$\max \sum_{i \in I} \sum_{j \in J} \sum_{l \in L} \delta_{i,j,l} \cdot \beta_{i,j,l} \cdot \gamma_{l,i} \cdot x_{i,j}, \tag{12}$$

subject to constrains (5), (6) and (11).

4 NUMERICAL EXPERIMENTS

To quantify the advantages of the proposed model, numerical experiments are performed. The computer programs for the optimization model are written in python and solved by IBM ILOG CPLEX 20.1.0 [17]. All computational tests are performed on a MacBook Pro equipped with an Intel 2.6 GHz CPU with 16 GB memory.

4.1 Dataset

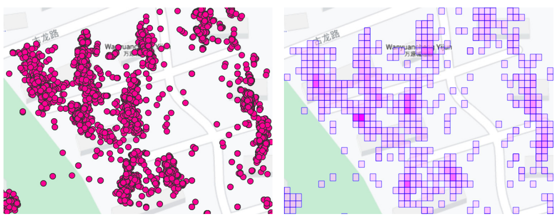
Meituan, the offline-to-online (O2O) specialist, was founded in 2010, and now is one of the world's largest online food delivery platforms. It had 290 million monthly active users and around 600 million registered users as of April 2018 [27]. In this study, OFD data are collected from 4 Chinese cities for 3 months from October 1st to December 31st 2021 by Meituan platform. A description and example of the dataset used in this study is shown in Table 1. For every order, it contains detailed GPS coordinates (fields of *user_lon* and *user_lat*) and address (the field of *user_addr*) of the user's location.

For the improved optimization model, before feeding the geospatial data into models, GPS points have to be converted into geohash cells. We use an example to illustrate the process, which is shown in Figure 6. Figure 6(a) illustrates original GPS points of orders, and the corresponding geohash cells are shown in Figure 6(b). From Figure 6(b), the different colors of geohash cells indicate numbers

Table 1: Major data types of an order in Meituan

Field	Description	Example
order_id	Unique identification of the order.	16015325888
user_id	Unique identification of the user.	320597876
user_lon	Longitude of the user's location.	121.397207
user_lat	Latitude of the user's location.	31.147946
user_addr	Text string of the user's address.	Room-N, XX
rider_id	Unique identification of the rider.	14790342

of orders located in them, and the geohash cell with darker color means it contains more orders. We treat every geohash cell as a GPS point, then feed them into single-AOI detection models.



(a) Original GPS points (b) Geohash cells

Figure 6: Visualization of geohash cells.

Due to the limitation of variables and constraints of CPLEX, instead of implementing the optimization model throughout the whole city, we partition the city into multiple regions (also called large grids) based on road network and implement the model in every region separately. Road network data of these 4 cities are collected, and every road segment is stored as a polyline, which consists of a sequence of geographic coordinates. Every road segment has a property of road hierarchy, and the road network partition can be implemented by the road hierarchy, which is shown in Figure 7. Figure 7(a) illustrates road segments based on hierarchy 1-5. Then the road network help us partition the urban area into multiple regions, as shown in Figure 7(b) [37]. In this figure, we use different colors to distinguish these regions. For every region, the shape is around 1km × 1km, and there are about 5 to 10 AOIs in it.

4.2 System Framework

The detailed system framework of the proposed approach is elaborated in Figure 8, consisting of four components:

Dataset. This component includes two types of data: (1) *Historical Order Data*, which include detailed addresses and GPS coordinates of customers; (2) *Road Network Data*, which include detailed information of every road segment.

Data Pre-processing. This component takes dataset of historical order and road network, then performs 6 main tasks: (1) *Hierarchy-Based Road Network Partition*, which partitions the road network into large grids based on road hierarchy; (2) *Large Grids*, which are the partitioned regions and OFD data located in the specific large grid are captured; (3) *Address Resolution*, which extracts AOI names and geographic coordinates of every order in the

(a) Road segments (b) Hierachy-based region partition

Figure 7: Road network and small partitioned regions in Wangjing area of Beijing.

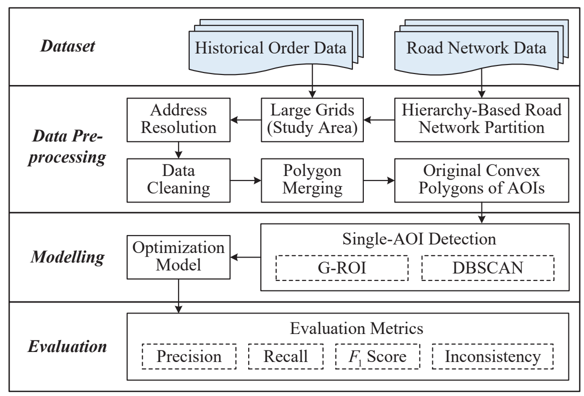


Figure 8: The detailed Framework of the proposed approach.

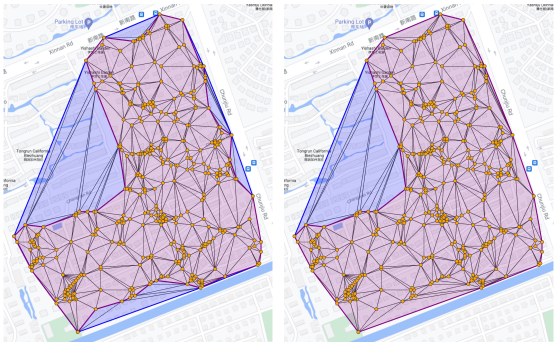
large grid; (4) *Data Cleaning*, which removes outlier GPS points; (5) *Polygon Merging*, which merges the same AOI with alias names; (6) *Original Convex Polygons of AOIs*, which constructs original boundaries of AOIs based on the convex hull.

Modelling. This component takes the preprocessed data as inputs, and generates boundaries of multi-AOIs, and then are fed into the optimization model. It includes two steps: (1) *Single-AOI Detection*, which generate a set of candidate spatial boundaries of every AOI based on algorithms of G-ROI and DBSCAN (detailed in Section 4.3); (2) *Optimization Model*, which uses the optimization model to detect multi-AOIs simultaneously based on the candidate spatial boundaries created in the previous step.

Evaluation. This component takes results of the optimization model, and evaluates the performance of results based on four metrics, which are precision, recall, F_1 score and inconsistency (detailed in Section 4.5).

4.3 Baseline Algorithms

In this section, we briefly review the baseline algorithms used in our numerical experiments. According to existing studies in single-AOI detection, G-ROI algorithm [4] achieves the best F_1 score compared with other methods, and DBSCAN algorithm [16, 33] has been widely used in AOI detection problem and achieves the robust performance. Additionally, we construct spatial boundaries of point



(a) Original alpha-shape (b) Modified alpha-shape

Figure 9: Visualization of alpha-shape algorithms. (a) and (b) show the boundaries (purple) by the original and modified alpha-shape algorithms, blue polygons are deleted parts.

clusters based on convex hull and modified alpha-shape concave hull.

4.3.1 G-ROI Algorithm. In [4], the authors propose a G-ROI algorithm for discovering ROIs on multiple social media datasets, and this algorithm achieves the best F_1 score among other detection methods. This algorithm contains two stages of reduction and selection. Let C be a set of geographic points within an AOI, and h_0 be the convex hull of C, and represented by a set of vertices.

The reduction stage begins from h_0, then it finds one of its vertices to generate the smallest polygon and remove this vertex. With the same strategy, it continues until the convex hull containing only three vertices. This stage returns a set of convex hulls $H = \{h_0, h_1, \cdots, h_n\}$ and a set of removed points $P = \{p_0, p_1, \cdots, p_{n-1}\}$ obtained through the n steps that have been processed. The selection stage tries to discover the cut-off point p_{cut}. In this study, we set different parameter values to construct candidate spatial boundaries of AOIs, and then feed them into the optimization model.

4.3.2 DBSCAN Algorithm. DBSCAN [13] is a density-based clustering algorithm, and is widely used in clustering for geospatial data. Compared with clustering methods like K-Means, DBSCAN does not require to specify the number of clusters, and can identify outlier points. Given a set of points on a Cartesian plane, DBSCAN can group together points with multiple nearby neighbors, and marks outlier points that lie alone in low-density regions (whose nearest neighbors are too far away). The working strategy behind DBSCAN is to identify the minimum number of neighboring points $minPts$ within the circle range of the radius ϵ. In comparison to other clustering algorithms, DBSCAN can identify clusters with different shapes and has good robustness for data noises [16].

Before using DBSCAN, the two parameters require to be set with proper values. The value of ϵ can be determined according to the geographic scale of the research problem. In general, if ϵ is set with a larger value, DBSCAN can construct AOIs with bigger coverage, while if the value is smaller, the produced AOIs are also smaller. The value of $minPts$ determines the minimum number of points of a cluster and represents the significance of the identified AOIs. If $minPts$ is set with a larger value, it can make sure to extract AOIs with a higher significance but may also miss some useful areas. while if $minPts$ is set with a smaller value, more clusters can be extracted but may also contain noisy points. In regard of the above reasons, it's difficult to find the best parameters of DBSCAN for every case, therefore, we set several groups of parameters to create different candidate spatial boundaries of AOIs, and then feed them into the optimization model.

4.3.3 Concave Hull. After identifying clusters of customers' locations, the next step is to construct polygons from these GPS points. The convex hull is a typical way to represent the external polygon of those points, and has been applied in many studies [14, 16]. While in some cases, the convex hull cannot match the boundary better but contains empty parts which do not belong to the original points. In our research, we try to find spatial boundaries of AOIs with no overlaps among each other, therefore, convex polygons cannot represent these AOIs properly. For more precise delineation of cluster shapes, Edelsbrunner *et al.* [12] propose the algorithm of alpha-shape, which can be used to construct a concave hull and represents the shape of the AOI more properly.

The steps for concave hull computation can be summarized as follows: (1) Generate a triangulated irregular network (TIN) of a set of discrete points using Delaunay triangulation method; (2) Remove exterior edges of the triangle with circumradius longer than a pre-defined length parameter l; (3) Repeat step 2 until every triangle's circumradius in the TIN is shorter than l; (4) Generate the resulted shape, which is the purple polygon in Figure 9(a). The parameter l can equal to any positive number. If l is less than the shortest circumradius r_{min}, then all edges are removed. If l is more than the longest circumradius r_{max}, then all edges are kept, and the generated hull will be the convex hull of the points. Therefore, only values between r_{min} and r_{max} can be the optimal value.

In this research, we modify the algorithm of alpha-shape to be more accurate for our dataset. For the algorithm of alpha-shape, it aims to delete all exterior edges with circumradius longer than l, therefore, it will create some empty parts. For the modified alpha-shape, it can create smoother boundaries, as described in Algorithm 1. For the input of this algorithm, $points$ means GPS points, and per refers to the percentage of deleted triangles. We first calculate $triangularMesh$ by Delaunay triangulation method. Then, the cicumradius of every triangle is calculated and sorted from large to small. A threshold of circumradius is calculated by per, we keep all triangles with the circumradius shorter than the threshold. For other triangles, we use the method of $cascaded\_union$ to merge adjacent triangles together, therefore some separated polygons are created, which are blue polygons in Figure 9(a). We calculate areas of these polygons, and find the largest and second largest ones $area1$ and $area2$. If $area1/area2$ is larger than 3, only the largest polygons are deleted, which is the blue polygon in Figure 9(b), and other parts are merged together as the return concave hull, which is the purple polygon in Figure 9(b). Otherwise, the convex hull of $points$ is returned.

4.4 Fine-tune Detected Boundaries

4.4.1 HMM Matching. In [29], the authors propose a map matching method by using HMM to discover the most probable route

Algorithm 1: Modified alpha-shape

```
Input: points, per
Output: concaveHull
begin
    triangularMesh ← DelaunayTriangulation(points)
    for (i_a, i_b, i_c) ∈ triangularMesh.vertices do
        circumR ← circumradius(a, b, c)
        edgePointsAppend(edgePoints, points[[i_a, i_b, i_c]])
        circumRAppend(circumRList, points[[i_a, i_b, i_c]])
    end
    circumRList ← sorted(circumRList)
    index ← round(per × len(circumRList)
    threshold ← circumRList[index]
    for i ← 0 to len(circumRList) do
        if circumRList[i] > threshold then
            deletedEdgePoints.append(edgePoints.pop(i))
        end
    end
    keptPolys ← cascaded_union(edgePoints)
    deletedPolys ← cascaded_union(deletedEdgePoints)
    concaveHull ← Polygon()
    if type(deletedPolys) = MultiPolygon then
        for poly ∈ deletedPolys do
            areaPolys.append(area(poly))
        end
        index_1, area_1 ← getIndexArea(areaPolys, 1)
        index_2, area_2 ← getIndexArea(areaPolys, 2)
        if area_1 / area_2 > 3.0 then
            deletedPoly ← deletedPolys.pop(index_1)
        end
    end
    concaveHull ← keptPolys ∪ deletedPolys
    return concaveHull
end
```

represented by a time series of coordinates, and achieve a good performance. In this study, every candidate spatial boundary has vertices of GPS points, which can be applied to this algorithm. This algorithm can be used to fine-tune the detected boundaries to match road network. Since vertices of AOIs' actual boundaries are not necessarily positioned on the road network, therefore, it does not perform good enough in cases without high quality internal road network. In this study, we use two indices to evaluate results, which are defined as:

$$Ratio_1 = \frac{Area\left(AOI_{found}\right)}{Area\left(AOI_{found} \cup AOI_{HMM}\right)}, \tag{13}$$

$$Ratio_2 = \frac{Area\left(MRR\left(AOI_{found}\right)\right)}{Area\left(MRR\left(AOI_{found} \cup AOI_{HMM}\right)\right)}, \tag{14}$$

where AOI_{found} is the detected AOI's boundary; AOI_{HMM} is the boundary calculated by HMM matching; $MRR(\cdot)$ is a function to get the minimum rotated rectangle of a polygon. And only if both two ratios are larger than the threshold, the HMM matching result will be accepted, otherwise will keep the original detected AOI's boundaries.

4.4.2 Grid Matching. We partition every research region into small grids by the internal road network, then match small grids to AOIs. We determine which AOI do these grids belong to by the ratio defined as follows:

$$Ratio = \frac{Area\left(AOI_{found} \cap Polygon_{Grid}\right)}{Area\left(Polygon_{Grid}\right)}, \tag{15}$$

where $Polygon_{Grid}$ is the polygon of every small grid. Then, we merge all small grids belonging to the same AOI, and use strategies mentioned in Section 4.4.1 to determine whether to keep the result or not.

Finally, these two fine-tuning algorithms are combined together to fine-tune the detected spatial boundaries. We compare the two results of HMM matching and grid matching, and keep the one with bigger area as the final result of the combining result.

4.5 Performance Metrics

Metrics of precision and recall are used to evaluate the performance of the baseline algorithms as well as our approach in detecting AOIs. The ground-truth AOIs in this study are manually labeled by ground survey. As in [4], let G_AOI_i be one of the ground-truth AOIs in a region, where $i \in I$, and let F_AOI_i be the corresponding found AOI by a single AOI detection method. Let $G\_AOI_i \cap F\_AOI_i$ be the corresponding true positive area, which is defined as the overlap of G_AOI_i and F_AOI_i. The two metrics are defined as:

$$\text{Precision} = \frac{\sum_{i\in I} Area(G\_AOI_i \cap F\_AOI_i)}{\sum_{i\in I} Area(F\_AOI_i)}, \tag{16}$$

$$\text{Recall} = \frac{\sum_{i\in I} Area(G\_AOI_i \cap F\_AOI_i)}{\sum_{i\in I} Area(G\_AOI_i)}, \tag{17}$$

where $\sum_{i\in I} Area(G\_AOI_i)$ is the whole areas of the ground-truth AOIs, and $\sum_{i\in I} Area(F\_AOI_i)$ is the whole areas of the corresponding found AOIs, and $\sum_{i\in I} Area(G\_AOI_i \cap F\_AOI_i)$ refers to the whole true positive areas in the region.

To sort the results, the F_1 score is defined as the harmonic mean of precision and recall as follows:

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{18}$$

To evaluate the overlap degree among different detected AOIs. We define the metric of inconsistency to calculate the overlap ratio of AOIs within a region as follows:

$$\text{Inconsistency} = \frac{Area\left(\cap_{i\in I} F\_AOI_i\right)}{Area\left(\cup_{i\in I} F\_AOI_i\right)}, \tag{19}$$

where $Area\left(\cap_{i\in I} F\_AOI_i\right)$ is the overlap area of all AOIs , and $Area\left(\cup_{i\in I} F\_AOI_i\right)$ is the union area of all AOIs in the region.

If the inconsistency is 0, it means all of those polygons are completely separated from each other. Higher inconsistency means a worse performance in this metric.

4.6 Optimization Results

In this study, we choose the densely populated area filled with large numbers of residential complexes as the research regions. Based on this principle, we choose 28 research cases from 4 Chinese cities. Two optimization models mentioned in Section 3 are represented as OM-1 (detailed in Section 3.2) and OM-2 (detailed in Section 3.3), which are applied to these research regions. After testing, two models achieve the same results on multi-AOIs detection, but they differ greatly in problem size. We report the summary statistics for OM-1 and OM-2 in Table 2. On average, they have the same number of binary variables, which is 199, but OM-1 has 13,017 constraints and OM-2 has only 1,525 constraints. We notice that instances can be solved in 78.7s and 13.8s by OM-1 and OM-2, respectively. Therefore, OM-2 has a significant efficiency improvement compared with OM-1.

Table 2: Optimization summary.

Design	Problem size		CPU time(s)
	Binary variables	Constraints	
OM-1	199	13,017	78.7
OM-2	199	1,525	13.8

Then we evaluate the average metrics of these research regions. Table 3 illustrates the performance (precision, recall, F_1 score, and inconsistency) of all single-AOI detection models, and the best values of the whole models are shown on the last row. G-ROI-01, G-ROI-02, $\cdots$, G-ROI-08 are models generated by G-ROI algorithm, and AVG-G-ROI refers to the average values of these 8 models; DBSCAN-01, DBSCAN-02, $\cdots$, DBSCAN-16 are models generated by DBSCAN algorithm, and AVG-DBSCAN represents the average values of these 16 models.

Results of the Table 3 show that the best precision of the whole single-AOI detection models is achieved by G-ROI, which is 0.983, while recall of it is the lowest one. The reason is that G-ROI algorithm removes too many points, therefore, in most cases, the ground-truth AOIs contain the found ones. DBSCAN achieves relatively high results (F_1 score ranging from 0.850 to 0.869), and the best recall and F_1 score of the whole single-AOI detection models is achieved by DBSCAN. On average of all models by DBSCAN, the precision is 0.895 and the recall is 0.825, which results in the F_1 score of 0.858. The precision is bigger than the recall, which means the identified AOIs are smaller than the ground-truth ones.

Table 4 illustrates the evaluation results of convex hull, optimization model, and fine-tuning algorithms. For the original convex hull, the precision and recall are 0.827 and 0.865, respectively, which results in the F_1 score of 0.847. The fact that the value of precision is lower than recall, means that the original convex hulls are on average bigger than the ground-truth ones. The inconsistency is 0.105, which means those AOIs have quite a lot of overlaps.

Then, the optimization model outperforms the other single-AOI detection algorithms in Table 3. The precision is 0.923 and the recall is 0.843, which leads to a F_1 score of 0.881. These results support the ability of the optimization model to discover multi-AOIs simultaneously. The inconsistency is 0, which means that the spatial boundaries of different AOIs are completely separated.

Table 3: Average precision, recall, F_1 score, and inconsistency of all single-AOI detection models in all cases

Model	Precision	Recall	F_1	Inconsistency
G-ROI-01	0.878	0.836	0.856	0.063
G-ROI-02	0.897	0.824	0.859	0.049
G-ROI-03	0.918	0.810	0.861	0.031
G-ROI-04	0.931	0.788	0.854	0.024
G-ROI-05	0.949	0.758	0.843	0.014
G-ROI-06	0.965	0.705	0.815	0.005
G-ROI-07	0.976	0.641	0.774	0.001
G-ROI-08	0.983	0.511	0.672	0.000
AVG-G-ROI	0.937	0.734	0.817	0.023
DBSCAN-01	0.851	0.865	0.858	0.080
DBSCAN-02	0.857	0.865	0.861	0.072
DBSCAN-03	0.887	0.851	0.869	0.038
DBSCAN-04	0.888	0.846	0.866	0.036
DBSCAN-05	0.889	0.844	0.866	0.036
DBSCAN-06	0.917	0.803	0.856	0.014
DBSCAN-07	0.918	0.801	0.856	0.014
DBSCAN-08	0.919	0.800	0.855	0.012
DBSCAN-09	0.858	0.838	0.847	0.048
DBSCAN-10	0.863	0.836	0.850	0.041
DBSCAN-11	0.900	0.837	0.867	0.022
DBSCAN-12	0.900	0.832	0.865	0.021
DBSCAN-13	0.901	0.830	0.864	0.021
DBSCAN-14	0.923	0.787	0.849	0.008
DBSCAN-15	0.924	0.786	0.850	0.008
DBSCAN-16	0.925	0.785	0.849	0.007
AVG-DBSCAN	0.895	0.825	0.858	0.030
Best values	0.983	0.865	0.869	0.000

Finally, algorithms of HMM and grid matching are applied to fine-tune the detected spatial boundaries of AOIs based on the road network data. After improved by the HMM algorithm, F_1 score increases to 0.892, and this value is 0.892 based on the grid matching algorithm. For the algorithm combining both HMM and grid matching, the best F_1 score is achieved, and the value is 0.894.

5 CONCLUSION AND FUTURE DIRECTIONS

In this paper, a novel optimization model is proposed to detect multi-AOIs simultaneously. Results of numerical experiments suggest that the detection results are promising, and our model achieves the best average F_1 score of the whole single-AOI detection models. To the best of our knowledge, this is the first instance of multi-AOIs detection. Moreover, we use two algorithms to fine-tune the detected spatial boundaries based on road network and achieve better performance of F_1 score.

Future work of this study includes: First, we only use two types of algorithms to create candidate spatial boundaries, while we can add more promising algorithms in the single-AOI detection step, which may help to improve the detection performance. Second, collecting more information of lower-level road network, which can

Table 4: Average precision, recall, F_1 score, and inconsistency achieved by convex hull, the optimization model, and fine-tuning algorithms in all cases

Model	Precision	Recall	F_1	Inconsistency
Convex Hull	0.827	0.868	0.847	0.105
Our Model	0.923	0.843	0.881	0.000
HMM Matching	0.888	0.896	0.892	0.000
Grid Matching	0.899	0.886	0.892	0.000
HMM+Grid	0.892	0.895	0.894	0.000

increase the possibility of further region partition. Third, since some AOIs are separated by rivers, walls or other man-made barriers, adding these networks can efficiently complement the road network. Fourth, a possible extension of this study is to add other data sources, such as remote sensing data, which can be used to perform image recognition of AOIs.

ACKNOWLEDGMENTS

This research was supported by Tsinghua University - Meituan Joint Institute for Digital Life.

REFERENCES

[1] Albino Altomare, Eugenio Cesario, Carmela Comito, Fabrizio Marozzo, and Domenico Talia. 2016. Trajectory pattern mining for urban computing in the cloud. *IEEE Transactions on Parallel and Distributed Systems* 28, 2 (2016), 586–599.

[2] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record* 28, 2 (1999), 49–60.

[3] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. 1996. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)* 22, 4 (1996), 469–483.

[4] Loris Belcastro, Fabrizio Marozzo, Domenico Talia, and Paolo Trunfio. 2018. G-roi: Automatic region-of-interest detection driven by geotagged social media data. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12, 3 (2018), 1–22.

[5] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 160–172.

[6] Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Sui. 2011. Exploring millions of footprints in location sharing services. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 5.

[7] Dorin Comaniciu and Peter Meer. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence* 24, 5 (2002), 603–619.

[8] David J Crandall, Lars Backstrom, Daniel Huttenlocher, and Jon Kleinberg. 2009. Mapping the world's photos. In *Proceedings of the 18th international conference on World wide web*. 761–770.

[9] Bidur Devkota, Hiroyuki Miyazaki, Apichon Witayangkurn, and Sohee Minsun Kim. 2019. Using Volunteered Geographic Information and Nighttime Light Remote Sensing Data to Identify Tourism Areas of Interest. *Sustainability* 11, 17 (2019), 4718.

[10] Alexandre Dubray, Guillaume Derval, Siegfried Nijssen, and Pierre Schaus. 2020. Mining Constrained Regions of Interest: An Optimization Approach. In *International Conference on Discovery Science*. Springer, 630–644.

[11] Matt Duckham, Lars Kulik, Mike Worboys, and Antony Galton. 2008. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern recognition* 41, 10 (2008), 3224–3236.

[12] Herbert Edelsbrunner, David Kirkpatrick, and Raimund Seidel. 1983. On the shape of a set of points in the plane. *IEEE Transactions on information theory* 29, 4 (1983), 551–559.

[13] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.

[14] Tianhui Fan, Naijing Guo, and Yujie Ren. 2021. Consumer clusters detection with geo-tagged social network data using DBSCAN algorithm: a case study of the Pearl River Delta in China. *GeoJournal* 86, 1 (2021), 317–337.

[15] Michael F Goodchild. 2007. Citizens as sensors: the world of volunteered geography. *GeoJournal* 69, 4 (2007), 211–221.

[16] Yingjie Hu, Song Gao, Krzysztof Janowicz, Bailang Yu, Wenwen Li, and Sathya Prasad. 2015. Extracting and understanding urban areas of interest using geotagged photos. *Computers, Environment and Urban Systems* 54 (2015), 240–254.

[17] IBM. 2021. Python tutorial of IBM ILOG CPLEX 20.1.0. Retrieved May 12, 2021 from https://www.ibm.com/docs/en/icos/20.1.0?topic=tutorials-python-tutorial

[18] Mohamed Ibrahim. 2020. *Extracting and mapping areas of interest from social media.* Ph. D. Dissertation. Wien.

[19] Lyndon S Kennedy and Mor Naaman. 2008. Generating diverse and representative image search results for landmarks. In *Proceedings of the 17th international conference on World Wide Web*. 297–306.

[20] Jastra Kranjec. 2021. Online Food Delivery Market to Hit $151.5B in Revenue and 1.6B users in 2021, a 10% Jump in a Year. Retrieved May 12, 2021 from https://stockapps.com/blog/2021/01/05/online-food-delivery-market-to-hit-151-5b-in-revenue-and-1-6bn-users-in-2021-a-10-jump-in-a-year

[21] Chiao-Ling Kuo, Ta-Chien Chan, I Fan, Alexander Zipf, et al. 2018. Efficient method for POI/ROI discovery using Flickr geotagged photos. *ISPRS International Journal of Geo-Information* 7, 3 (2018), 121.

[22] Dmitry Laptev, Alexey Tikhonov, Pavel Serdyukov, and Gleb Gusev. 2014. Parameter-free discovery and recommendation of areas-of-interest. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 113–122.

[23] Charlene Li, Miranda Mirosa, and Phil Bremer. 2020. Review of Online Food Delivery Platforms and their Impacts on Sustainability. *Sustainability* 12, 14 (2020), 5528.

[24] Linna Li and Michael F Goodchild. 2012. Constructing places from spatial footprints. In *Proceedings of the 1st ACM SIGSPATIAL international workshop on crowdsourced and volunteered geographic information*. 15–21.

[25] Jiajun Liu, Zi Huang, Lei Chen, Heng Tao Shen, and Zhixian Yan. 2012. Discovering areas of interest with geo-tagged images and check-ins. In *Proceedings of the 20th ACM international conference on Multimedia*. 589–598.

[26] Xinyi Liu, Qunying Huang, and Song Gao. 2019. Exploring the uncertainty of activity zone detection using digital footprints with multi-scaled DBSCAN. *International Journal of Geographical Information Science* 33, 6 (2019), 1196–1223.

[27] Meituan. 2021. Meituan Announces Financial Results for the Year Ended December 31, 2020. Retrieved June 8, 2021 from https://www.prnewswire.com/news-releases/meituan-announces-financial-results-for-the-year-ended-december-31-2020-301256643.html

[28] Seyed Morteza Mousavi, Aaron Harwood, Shanika Karunasekera, and Mojtaba Maghrebi. 2017. Geometry of interest (GOI): spatio-temporal destination extraction and partitioning in GPS trajectory data. *Journal of Ambient Intelligence and Humanized Computing* 8, 3 (2017), 419–434.

[29] Paul Newson and John Krumm. 2009. Hidden Markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. 336–343.

[30] Carlos Ruiz, Myra Spiliopoulou, and Ernestina Menasalvas. 2007. C-dbscan: Density-based clustering with constraints. In *International workshop on rough sets, fuzzy sets, data mining, and granular-soft computing*. Springer, 216–223.

[31] Motohiro Shirai, Masaharu Hirota, Hiroshi Ishikawa, and Shohei Yokoyama. 2013. A method of Area of Interest and Shooting Spot Detection using Geo-tagged Photographs. (2013).

[32] Evaggelos Spyrou and Phivos Mylonas. 2016. Analyzing Flickr metadata to extract location-based information and semantically organize its photo content. *Neurocomputing* 172 (2016), 114–133.

[33] Vishal Srivastava, Priyam Tejaswin, Lucky Dhakad, Mohit Kumar, and Amar Dani. 2020. A Geocoding Framework Powered by Delivery Data. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*. 568–577.

[34] Daniel Sui, Sarah Elwood, and Michael Goodchild. 2012. *Crowdsourcing geographic knowledge: volunteered geographic information (VGI) in theory and practice.* Springer Science & Business Media.

[35] Junjie Sun, Tomoki Kinoue, and Qiang Ma. 2020. A City Adaptive Clustering Framework for Discovering POIs with Different Granularities. In *International Conference on Database and Expert Systems Applications*. Springer, 425–434.

[36] Huy Quan Vu, Gang Li, Rob Law, and Ben Haobin Ye. 2015. Exploring the travel behaviors of inbound tourists to Hong Kong using geotagged photos. *Tourism Management* 46 (2015), 222–232.

[37] Nicholas Jing Yuan, Yu Zheng, and Xing Xie. 2012. Segmentation of urban areas using road networks. *Microsoft, Albuquerque, NM, USA, Tech. Rep. MSR-TR-2012-65* (2012).

[38] Libao Zhang and Shiyi Wang. 2017. Region-of-interest extraction based on local–global contrast analysis and intra-spectrum information distribution estimation for remote sensing images. *Remote Sensing* 9, 6 (2017), 597.

[39] Jie Zheng, Ling Wang, Shengyao Wang, Yile Liang, and Jize Pan. 2021. Solving two-stage stochastic route-planning problem in milliseconds via end-to-end deep learning. *Complex & Intelligent Systems* 7, 3 (2021), 1207–1222.