Identifying Points of Interest using Heterogeneous Features

YIYANG YANG, University of Macau
ZHIGUO GONG, University of Macau
LEONG HOU U, University of Macau

Deducing trip related information from web-scale datasets has received large amounts of attention recently. Identifying points of interest (POIs) in geo-tagged photos is one of these problems. The problem can be viewed as a standard clustering problem of partitioning two dimensional objects. In this work, we study spectral clustering which is the first attempt for the POIs identification. However, there is no unified approach to assign the subjective clustering parameters; especially these parameters are immensely varying in different metropolitans and locations. To address this issue, we are intent to study a self-tuning technique which can properly determine the parameters for the clustering needed. Besides geographical information, web photos inherently store other rich information. Such heterogeneous information can be used to enhance the identification accuracy. Thereby, we study a novel refinement framework which is based on the tightness and cohesion degree of the additional information. At last, we thoroughly demonstrate our findings by web scale datasets collected from Flickr.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Clustering; H.2.8 [Database Applications]: Data Mining, Image Databases, Spatial Databases and GIS

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Web Images, Spectral Clustering

ACM Reference Format:
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

Nowadays people are used to sharing diverse type of resources with others via social network services. Web album is one of the examples where people can share their photos with others. Typically, the web photos contain diverse information, such as taken time, location, descriptive tags and so on. Such large amounts of user-contributed data consist of spatial, temporal, textual, and visual information that can be used for different mining tasks. Deducing trip related information from web photos [Serdyukov et al. 2009] has been an active topic recently. These works include mapping the photos [Crandall et al. 2009; Serdyukov et al. 2009], identifying places of interest [Kennedy and Naaman 2008; Kisilevich et al. 2010], travel movement mining [Zheng et al. 2012], predicting user travel behavior [Clements et al. 2010], itinerary mining [Popescu and
Grefenstette 2009; Popescu et al. 2009; Jain et al. 2010], itinerary planning [Choudhury et al. 2010], and so on.

In this work, we consider the collection of photos from Flickr that provides rich spatial (e.g., geographical location), temporal (e.g., taken time), and textual (e.g., photo tags) information. The geographical location was automatically captured by location aware mobile devices or manually inputted by geo-tagging tool. The taken time was automatically recorded into the photo meta-data. The photo tags were manually created by users who uploaded the photos. There are currently over 178 million geo-tagged photos on Flickr while there were only 40 million as reported by [Kennedy and Naaman 2008] in 2008. With the advent of more location aware devices, there is no doubt that the number of geo-tagged photos on Flickr grows at a rapid pace. These result in substantial research for web scale mining processes.

Our goal in this work is to identify high quality points of interest (POIs) using the collection of geo-, time- and text-tagged photos. There are some results recently reported in the community. Basically, all these works are based on an observation that people often take more photos in the hot area of the points of interest (POIs). Kennedy and Naaman [Kennedy and Naaman 2008] identify POIs in a city by applying k-means clustering on geo-tagged photos. The identified POIs are subsequently ranked by their textual and visual features. Crandall et al. [Crandall et al. 2009] study a similar problem on finding POIs in web scale datasets which outperforms Kennedy and Naaman [Kennedy and Naaman 2008] using mean shift clustering [Comaniciu and Meer 2002]. However, their work requires a parameter density radius for the clustering process which is set by their subjective observation (e.g., 100 meters). We argue that such radius setting is difficult to set subjectively since the size of POIs varies immensely in different metropolitans and locations. For instance, the size of famous POIs, such as Eiffel Tower, Musée du Louvre, and Arc de Triomphe, in Paris is quite different as illustrated in Fig. 1. Besides k-means and mean shift clusterings, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [Ester et al. 1996] is recently studied for POIs identification in [Kisilevich et al. 2010] and [Zheng et al. 2012]. However, DBSCAN suffers from similar drawback which requires to set subjective radius parameters. Besides the parameter issue, there is limited work to consider heterogenous information in the POIs identification process. To the best of our knowledge, migrating heterogenous information into the identification process is still an open problem which is worth to further exploit.

Inspired by the above discussion, we have two main missions in this work, including (1) avoiding using subjective parameters in clustering and (2) bringing heterogenous information into the clustering processing. The first mission is done in our previous work [Yang et al. 2011] which avoids using any subjective clustering parameter from the clustering process. We call our solution a self-tuning clustering method since it can identify POIs neither knowing (i) the number of POIs, (ii) the size of POIs, nor (iii) the shape of POIs in advance. In particular, the self-tuning clustering is based on spectral clustering [Bühler and Hein 2009; Shi and Malik 2000] that is a widely accepted clustering method in image segmentation. Given a graph with edge weights (corresponding to the spatial distance of the objects), the spectral clustering attempts to cut the graph into a set of subgraphs by the spectrum of the graph similarity such that the effort of the graph decomposition is minimized. One of the spectral clustering models is to recursively bi-partition the graph subject to a termination criteria, such as total number of subgraphs or the robustness of the subgraphs. However, as discussed in the example (Fig. 1), the size of POIs is not necessarily identical in a city. There is no unified termination criteria for the POIs in different metropolitans or locations. Zelnik-Manor and Perona [Zelnik-Manor and Perona 2004] propose a spectral clustering that can automatically decide the number of clusters based on eigenvector analysis [Zelnik-Manor...
Identifying Points of Interest using Heterogenous Features

Fig. 1: Famous POIs in Paris

and Perona 2004; Xiang and Gong 2008]. However, their techniques prefer to return a result having small number of clusters since the eigenvector analysis considers the data in its entirety but ignores their local effect. In our experimental testings, the clustering approach of [Zelnik-Manor and Perona 2004] only partitions the Paris dataset into 4 clusters which is clearly inapplicable for POIs identification.

Thus, we attempt to make the spectral clustering self-tuned (i.e., automatically deciding the number of clusters with no subjective parameter) based on the spatial distribution of the photos. This is inspired by an observation that the geo-tagged photos are likely Gaussian distributed in surrounding areas of POIs. As a remark, this observation is used in many relevant work [Cooper 2011; Huang et al. 2012]. Our goal is to make the termination criteria not further partition a subgraph when the objects of the subgraph are Gaussian distributed. We find that if a subgraph $C$ is Gaussian distributed, then the cost to decompose $C$ and its subsequent partitions is similar. In other words, $C$ can be viewed as a POI if this cost pattern is detected. The detail of our finding and analysis is given in Section 3.2. In this manuscript, we compare with more well-known clustering techniques which further secures the robustness of our self-tuning method in POIs identification process.

The second mission is newly proposed in this manuscript which attempts to identify POIs by taking heterogeneous information (besides spatial information) into consideration. Intuitively, if the photos have high coherence in terms of temporal and textual information, then we should group them into a POI. However, there is no simple method to migrate such information into a self-tuning framework since the heterogeneous information may no longer follow the Gaussian distribution. Thereby, we study a two-step refinement framework where (1) we first partition the photos by our self-tuning technique based on spatial information and (2) then refine the clusters by the closeness degree of the self-tuning results in terms of other supportive information (e.g., temporal and textual information). For such sake, we define a pair of closeness metrics, tightness and cohesion, that assesses the inter-closeness of two clusters and the intra-closeness of a cluster, respectively. Our experimental study demonstrates that our refinement framework has significant improvement over our self-tuning technique.
We summarize our main contributions as follows.

1. We attempt to remove subjective parameter (e.g., radius) from the POIs identification process. This is important as the size and the number of POIs are immensely varying in different metropolitans and locations and there is no standard approach to define appropriate parameters.

2. We give a complete survey on POIs identification and thoroughly compare the state-of-the-art with our self-tuning POI identification using web scale datasets. In addition, we evaluate our refinement model and study its effectiveness in the experiments.

3. We verify the robustness of the self-tuning method by comparing with diverse well-known clustering methods, including $k$-means, kernel $k$-means, fuzzy $c$-means, P-DBSCAN, mean shift, and spectral clustering. Our self-tuning method outperforms these methods without tuning any parameter in advance.

4. We newly propose a two-step refinement framework based on the cohesion and tightness degree of the clusters. We refine the self-tuning result by taking other supportive information into consideration. More importantly, this refinement framework is also parameter-free which shares the same intuition of the self-tuning technique.

5. We take the textual information of photos (i.e., tags) into the refinement process, which further enhance the identification quality.

6. We analyze the time complexity of the self-tuning technique and experimentally evaluate the computation performance of the framework.

Note that the first two contributions are directly derived from our prior work [Yang et al. 2011]. In our previous work, the refinement process is done by a reinforcement framework which requires huge effort on tuning the parameters. To address the tuning issues, we study a completely new refinement framework which refines the self-tuning results by their closeness in terms of other supportive information. The new refinement framework not only secures the identification robustness but also makes our work towards a unified goal, i.e., self-tuning.

The rest of this paper is organized as follows. We first overview and compare the state of the art POIs identification in Section 2. Next, we discuss spectral clustering and our feature similarity measurement in Section 3. Our refinement model is introduced in Section 4. Our approaches are thoroughly evaluated in Section 6 using web scale datasets collected from Flickr. Before we conclude our work and discuss possible future work in Section 8, we also introduce some related work in deducing trip related information in Section 7.

2. CLUSTERING TECHNIQUES FOR POIS IDENTIFICATION

Identifying POIs from a collection of geo-tagged photos can be viewed as a clustering problem of identifying highly photographed locations. Kennedy and Naaman [Kennedy and Naaman 2008] use $k$-means clustering for POI identification with the collection of geo-tagged photos. $k$-means clustering aims to partition $n$ objects into $k$ clusters such that each object belongs to the closest cluster. Formally, the objective function of $k$-mean can be formulated as follows.

$$
\text{minimize} \sum_{1 \leq i \leq k} \sum_{o \in C_i} ||o - \mu_i||^2
$$

where $C_i$ is one of the $k$ clusters and $\mu_i$ is the centroid of $C_i$. Finding the exact solution of $k$-means is an NP-hard problem. The most common heuristic algorithm adopts an iterative refinement which alternates the cluster objects until the mean of each cluster becomes stable. However, the $k$-means clustering is a fixed-size clustering approach
Identifying Points of Interest using Heterogenous Features

which requires to give the number of clusters in advance. As discussed in Section 1, there is no appropriate rule to decide the value of \( k \) for different metropolitans and locations. Besides general \( k \)-means, there is a method, called fuzzy \( c \)-means clustering [Nock and Nielsen 2006], which attempts to enhance the clustering performance by fuzzy techniques. However, the fuzzy \( c \)-means clustering is still not a parametric-free technique. In the category of the mean-based clustering techniques, \( g \)-means clustering [Hamerly and Elkan 2003] and its variants can automatically determine the number of clusters for Gaussian distributed datasets. However, these clustering techniques are investigated in the context of image segmentation problems which has small number of clusters (e.g., 10 clusters) in general. Obviously this setting is not applicable for POI identification (i.e., there are more than 200 POIs in a city in general).

Crandall et al. [Crandall et al. 2009] study mean shift clustering [Comaniciu and Meer 2002] where the number of clusters (i.e., \( k \)) is not required. To identify the clustering results, the mean shift clustering locates the maxima by sampling discrete data based on a kernel function, where the kernel function \( f_K \) could be a typical Uniform kernel (i.e., \( f_U(o_N-o) = 1 \)) or Gaussian kernel (i.e., \( f_G(o_N-o) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\|o_N-o\|^2}{2\sigma^2}} \)). Accordingly, the mean is estimated by

\[
\mu = \frac{\sum_{o_N \in N(o)} f_K(o_N-o) o_N}{\sum_{o_N \in N(o)} f_K(o_N-o)}
\]

where \( N(o) \) is the neighborhood of \( o \) and \( \mu \) is the new mean estimated by the kernel function. However, the mean shift clustering is still not a parametric free technique as it requires to set a neighborhood bandwidth (i.e., \( N(o) \)). The clustering quality is substantially affected by the bandwidth which can be viewed as the influential area of an object.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [Ester et al. 1996] is another widely used clustering technique, which is based on the notion of density reachability. To identify the clustering results, DBSCAN groups the objects into a cluster if the objects are density reachable. More specifically, an object \( o \) is directly density reachable from another object \( o' \) if their distance does not exceed a given threshold \( \varepsilon \) and \( o' \) has more than \( \theta \) neighbor objects. \( o \) is called density reachable from \( o'' \) if there is a directly density reachable sequence of objects from \( o \) to \( o'' \). However, the reachability definition does not works well when the objects fall in skewed distribution (e.g., POI photos). Kisilevich et al. [Kisilevich et al. 2010] figure out this problem and propose P-DBSCAN which refines the density reachable definition by an adaptive technique. The only difference in P-DBSCAN is that the neighbor object requirement in the reachability definition is declined proportionally to the distance of \( o \) and \( o' \).

**Summary.** None of the above methods can identify high quality POIs without giving proper parameters or kernels. This makes the quality of the POI identification vary on different datasets if we directly adopt the solutions discussed above. A good POI identification solution should remove all subjective parameters (e.g., radius and neighborhood constraints) from the clustering process. Thus, we study a new POI identification technique in this work which can identify POIs from a collection of geo-tagged photos by a self-tuning process.

**3. CUT TECHNIQUES FOR POIS IDENTIFICATION**

In this section, we attempt to study another clustering approach, spectral clustering, for the POI identification problem (Section 3.1). To address the subjective parameters in the spectral clustering, we propose a self-tuning technique based on the partitioning costs along the entire clustering process (Section 3.2).
3.1. Spectral clustering

Spectral clustering [Bühler and Hein 2009; Cheeger 1970; Chung 1997; Shi and Malik 2000; Ng et al. 2001] has evolved into one of the most common clustering techniques and has been widely applied in many applications recently. Roughly speaking, the spectral clustering iteratively seeks a bipartition \(^1\) of an object collection until all sub-partitions fulfill a termination criteria. Seeking a bipartition can be done by various cut techniques, such as ratio cut [Wang and Siskind 2003], normalized cut [Hochbaum 2010; Shi and Malik 2000], and Cheeger cut [Bühler and Hein 2009; Cheeger 1970]. These techniques have been shown very effective in image segmentation [Shi and Malik 2000]. However, to the best of our knowledge, no previous work attempts to identify POIs by the spectral clustering techniques.

Table I: List of cut techniques

<table>
<thead>
<tr>
<th>Name</th>
<th>Objective</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio cut</td>
<td>( \min_{V_s \subset V} \frac{C(V_s, \bar{V}_s)}{</td>
<td>V_s</td>
</tr>
<tr>
<td>Normalized cut</td>
<td>( \min_{V_s \subset V} \frac{C(V_s, \bar{V}_s)}{C(V_s, V)} + \frac{C(V_s, \bar{V}_s)}{C(\bar{V}_s, V)} )</td>
<td>NP-hard</td>
</tr>
<tr>
<td>Cheeger cut</td>
<td>( \min_{V_s \subset V} \min {C(V_s, \bar{V}_s), C(\bar{V}_s, V)} )</td>
<td>NP-hard</td>
</tr>
</tbody>
</table>

Seeking a bipartition from an object collection is similar to finding a cut from a graph. In graph theory, a cut is to remove some edges from a graph \( G \) such that \( G \) is decomposed into two subgraphs subject to an objective function. The objective and complexity of three different cut techniques are summarized in Table I, where \( G = (V, E, W) \) is an undirected graph and \( C(V_s, \bar{V}_s) \) is the weight sum of the edges between \( V_s \) and \( \bar{V}_s \), i.e., \( C(V_s, \bar{V}_s) = \sum_{i \in V_s, j \in \bar{V}_s} w_{ij} \).

These cut problems can be transformed into a quadratic discrete optimization problem, Rayleigh problem, where the optimization objective is shown in Equation 3.

\[
\min_{y} y^T L y \quad \text{subject to} \quad y_i \in \{-b, 1\}, \quad y^T Q y \leq 1
\]

where \( y \) is a vector having \( n \) variables, \( L = W^D - W \) is referred as the Laplacian of the graph, \( W \) is referred as the weight matrix of the graph, \( W^D \) is a diagonal matrix with \( W^D_{ii} = \sum_{(i,j) \in E} W_{ij} \), and \( Q \) is a diagonal non-negative matrix that is set according to the selected cut technique. For example, \( Q \) is set to \( I \) in ratio cut and is set to \( W^D \) in normalized cut.

The Rayleigh’s optimization is a well known NP-hard problem. A common approach is to relax the integral constraints, such as replacing \( y_i \in \{-b, 1\} \) by \( y_i \in [-b, 1] \). In the solution of the graph cut problems, the relaxation is done by transforming the problem into an eigenvector problem. Every cut technique should have their corresponding eigenvector problem for the relaxation transformation. For the sake of readability, we only introduce the normalized cut in this manuscript and the detail of other transformations can be found in [von Luxburg 2007].

Algorithm 1 shows the spectral clustering framework using normalized cut. After preparing all necessary matrices in the first two steps, we can compute the second smallest eigenvalue \( \lambda \) and the corresponding eigenvector \( V \) by Lanczos algorithm [Golub and Van Loan 1996] in step 3. As a remark, step 3 is the only step to be replaced

\(^1\)There is other variants of spectral clustering which may seek for \( k \)-way partitions.
ALGORITHM 1: Spectral Clustering by Normalized Cut

**Input:** The weight Matrix of the Graph \( W \)

**Output:** The cluster assignment of the graph vertices \( V \)

1. Compute Diagonal Matrix \( W^D \)
2. Compute Laplacian matrix \( L \) by \( W \) and \( W^D \)
3. **Compute the second smallest eigenvalue** \( \lambda \) and the corresponding eigenvector \( \nu \) of eigenvector problem \( L\nu = \lambda D\nu \)
4. **if** the eigenvalue \( \lambda \) is smaller than threshold \( \delta \) **then**
   1. Decompose the graph into \( W_t \) and \( W_h \) according to \( \nu \)
   2. Call \( SC_{NC}(W_t) \) and \( SC_{NC}(W_h) \)
   **end**

if other cut technique is adopted. Based on the previous study [Shi and Malik 2000], the second smallest eigenvalue \( \lambda \) can be viewed as the minimum cost to decompose the graph subject to the corresponding objective function. The graph is decomposed into two if the cost \( \lambda \) is smaller than the threshold \( \delta \). The decomposition is done by picking a value \( v \) (e.g., medium) in the corresponding eigenvector \( \nu \). The elements in the eigenvector are accordingly split into two groups where the first group contains all elements having better value than \( v \) and the second group contains the remaining elements. Note that the selection of \( v \) is substantial to the decomposition. Thereby, we test multiple \( v \) and pick the best value (i.e., maximize the objective function) such that the quality of decomposition is maximized. Even though spectral clustering is shown to provide good result in many applications, it still requires a subjective parameter, \( \delta \), that is hard to define properly for different datasets.

**The effect of \( \delta \).** We demonstrate the parametric issues using a real dataset from Flickr. The dataset is a Paris's photos collection with 216,300 geo-tagged photos in total. The detail of our data preparation can be found in Section 6. All figures are generated by our modified version of Java OpenStreetMap Editor \(^2\). For the ease of presentation, we highlight the clusters by different colors such that the size and shape of the clusters are clearly illustrated. In addition, we also plot the minimum bounding rectangles (MBRs) of tourist attractions for reference, where the tourist attractions are collected from the metadata of OpenStreetMap \(^3\).

The trivial parameters of \( k \)-means, mean shift, and P-DBSCAN are set to their default values as reported in the state-of-the-art [Kennedy and Naaman 2008; Crandall et al. 2009; Kisilevich et al. 2010]. We carefully tune other parameters and generate the snapshots of the best results in terms of their \( F1 \) score (details can be found in Section 6). Their best settings are shown in the caption of Fig. 2. For instance, we set \( k = 100 \) for \( k \)-means.

According to the illustration in Fig. 2, some attraction areas are precisely identified by these clustering approaches. For instance, the world famous POIs, Eiffel Tower and Musée du Louvre, are identified very well by P-DBSCAN in Fig. 2(c) and spectral clustering in Fig. 2(d), respectively. By carefully tuning their parameters, these clustering approaches may provide even better identification quality. However, the tuning time may be very long and the tuning effort cannot transform to other metropolitans and locations. In summary, these clustering approaches are very sensitive to their subjective parameters, e.g., \( k \) (the number of clusters in \( k \)-means), \( \eta \) (influential area in mean shift), \( \varepsilon \) (density radius in DBSCAN), and \( \delta \) (cost threshold in spectral clustering). This

---

\(^2\)http://josm.openstreetmap.de/

\(^3\)http://www.openstreetmap.org/
confirms our claim that the subjective parameters should be removed from the POI identification process.

3.2. Self-Tuning Spectral Clustering

Inspired by the above discussion, we attempt to eliminate the effect of threshold $\delta$ from the spectral clustering such that the POI identification process becomes parametric free and self-tuned. In this work, we only focus on the spectral clustering technique since 1) the calculation (Algorithm 1) can be viewed as a hierarchical framework so that each iteration is trackable and 2) the decomposition cost (i.e., the second smallest eigenvalue $\lambda$) can be further exploited. These features are not available in other clustering approaches to the best of our knowledge.

Our self-tuning technique is based on an observation that the photos in a city are not uniformly distributed especially at the area nearby POIs. Fig. 3(a) and Fig. 3(b) plot the photos nearby Eiffel Tower and Musée du Louvre in Paris. In these examples, the photos are more likely in Gaussian distribution. In order to show the robustness of this observation, we extract several photo sets from different areas in Paris and New York. For every photo set, we verify their distribution by Anderson-Darling test [Hamerly and Elkan 2003], which is a statistical test of data distribution. More specifically, Anderson-Darling test calculates the statistic value $A^{\star 2}$ from the given photo set and
compare this value against the critical value of the theoretical distribution. If $A^{*2}$ of a photo set exceeds a given critical value, then the hypothesis of normality is rejected with some significance level. Otherwise, the set of photos more likely falls into Gaussian distribution.

We apply the Anderson-Darling test to verify the distribution of the photos from our photo collection. Fig. 4(a) plots the photos around a POI “Tour Montparnasse” in Paris. The $A^{*2}$ of this area is 0.762 which is much smaller than the critical value of the theoretical distribution, i.e., 1.869. Thus, the hypothesis of normality is accepted for this area of photos, which means the photos in this POI area are likely Gaussian distributed. As a remark, in majority of our testings, $A^{*2}$ is small when the input of the test is from an POI area. Fig. 4(b) demonstrates another example where the photos are drawn from a non-POI area in Paris. The $A^{*2}$ statistic in this area is 1.886 which is small.

\footnote{We set the level of significance $\alpha$ to 0.0001, which means that there is only 1% wrong decision among 100 testings. For the detail, please refer to [Hamerly and Elkan 2003].}
higher than the critical value. If we manually analyze the data distribution, this area of photos should be partitioned into two groups instead of a single Gaussian.

According to our discussion, the distance relationship, $D_{ij}$, should be modeled by a Gaussian equation as follows.

$$D_{ij} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{d(i,j)^2}{2\sigma^2}}$$

(4)

where $d(i,j)$ is the Euclidean distance in between photo $i$ and $j$ and $\sigma^2$ is the variance of the Gaussian function.

For the sake of our subsequent discussions, we illustrate two concrete sets of Gaussian distributed POI photos and the spectral clustering process in Fig.5(a) and Fig.5(b). Suppose we apply the spectral clustering using normalized cut on these photos, which seeks for a balanced partitions. The 1st cut is indicated by the solid line where the cut cost $\lambda = 0.55$. When setting the threshold $\delta = 0.6$, the clustering process perfectly identifies two POIs $A$ and $B$. However, when setting $\delta = 0.81$, the spectral clustering will execute one more cut such that the photos in $A$ are decomposed into 2 small clusters, $A_1$, $A_2$. This example clearly shows that the value of $\delta$ is substantial to the clustering process.

To remove the subjective parameter $\delta$ from the spectral clustering, we observe that the cost to decompose a dense region, $A (\lambda = 0.8)$, is similar to the cost to decompose its sub-clusters, e.g., $A_1 (\lambda = 0.82)$ and $A_2 (\lambda = 0.82)$. However, the cost to decompose $A$’s parent is much lower, $A \cup B (\lambda = 0.55)$. This is because $A$ and $A_1$ share similar distribution (i.e., Gaussian distribution) but $A \cup B$ is not. We summarize this finding in Observation 1.

**Observation 1.** Given the execution tree of a spectral clustering process, the cost to decompose a cluster $C$ and its sub-clusters is similar if the elements in $C$ are Gaussian distributed.

According to Observation 1, a cluster should be preserved in the result if the cost to cut its sub-clusters is similar. A straightforward metric is to measure the similarity between the cost to cut a cluster $C$ and its children sub-clusters. However, this simple metric is very fluctuated since it only assesses two consecutive clustering procedures.

For each photo set, we attempt to discover $k$ Gaussian components by the Expectation Maximization (EM) algorithm.
Therefore, we propose a new metric that assesses the cost similarity of clusters in a spectral clustering process. We first define a new concept, cut path $p$, as follows.

**Definition 3.1 (Cut Path).** Given the hierarchical view of a spectral clustering process, a cut path of a cluster $C$ is the path from $C$ to a leaf cluster.

Furthermore, we define the cost of a cut path $\text{cost}(p)$ that is the average cut cost along $p$.

$$\text{cost}(p) = \frac{\sum_{i \in p} \lambda_i}{\text{length}(p)}$$ (5)

where $\text{length}(p)$ indicates the length of the path $p$. We use our running example in Fig.5(b) to explain these two definitions. In this example, a valid cut path $p$ of $A \cup B$ is $A \cup B \rightarrow A \rightarrow A_2$, where the cost of this path is $(0.55+0.8+0.82)/3 = 0.72$.

**Definition 3.2 (Cut Cost Similarity).** Given the hierarchical structure of a spectral clustering, the similarity of a cluster $C$ is defined in

$$\text{sim}(C) = 1 - \left( \sum_{p \in P_C} \frac{|\text{cost}(p) - \mu_C|}{|P_C|} \right)$$ (6)

where $P_C$ is the set of cut paths under cluster $C$, $\mu_C$ is the mean cost of all cut paths in $P_C$, and $|P_C|$ indicates the number of paths in $P_C$.

Given the cost of every cut path, we can define the cost similarity of the paths under a cluster in the spectral clustering process (Definition 3.2). If the cost similarity of $C$ is high, then $C$ is more likely gaussian distributed according to Observation 1.

To verify the robustness of our observation, we plot the average cost similarity of POI areas and non-POI areas as a function of hierarchical levels in the spectral clustering process. In this experiment, we randomly draw 20 POI areas and 20 non-POI areas, respectively, from our Paris photo collection (cf. Section 6). Fig.6 clearly shows that the cost similarity of POI areas grows much faster than non-POI areas. Thereby, the claim in Observation 1 is reasonable where the cost of decomposing POI areas (i.e., Gaussian distributed objects) is more similar than non-POI areas.

Given the spectral clustering processing tree (cf. Fig.5(b)), our self-tuning technique is to discover high quality POIs using a top-down execution paradigm. A node in the execution tree is picked as a result $C$ when $\text{sim}(C)$ reaches a good local peak. In this
work, we simply set this value to a local peak that has the longest piecewise subsequence\(^6\), where the piecewise subsequence is composed of an non-decreasing subsequence and an non-increasing subsequence. As a concrete example, we selectively plot the cut similarity of an POI area in Fig.7. In this example, we pick the value at level 20 as the local peak since it has the longest piecewise subsequence, i.e., an non-decreasing subsequence from level 10 to 20 and an non-increasing subsequence from level 20 to 25. As a remark, the local peak can be viewed as the best point to reserve the cluster result which avoids wrongly partitioning POIs based on the cut cost similarity. The effectiveness of this approach is shown in Section 6.

In summary, our self-tuning technique uses two steps to identify high quality POIs, including (1) the spectral clustering and (2) the cut cost similarity assessment. The first step takes \(O(m^3 + nk(m + t))\) time [Chen et al. 2011], where \(n\) is the number of photos, \(k\) is the number of clusters, \(m = k \times \alpha\) (where \(\alpha\) is around 2 to 3 in practice), \(t\) is average number of affiliated edges of an object. Given the result (i.e., the execution tree) of the spectral clustering, the second step (i.e., calculating the cut cost similarity) can be viewed as a tree traversal from leaves to root, which takes \(O(k \log k)\) time. It is obvious that the first step dominates the computation cost.

4. REFINEMENT WITH MULTIPLE FEATURES

So far, the identification process only utilizes the spatial information. Apparently, there is room for improvement as web photos provide not only spatial information but also other rich metadata, such as temporal and textual information.

Let us consider a concrete example in Fig. 8, where we have 16 photos in the system. From geographical point of view, we might naturally form 2 clusters where the left cluster (in blue) contains 12 photos and the right cluster (in gray) contains the remaining 4 photos. If temporal information is taken into account, we may have different result. In the temporal space, the arrows between the photos represent the movement sequence of photographers. By considering both spatial distances and movement sequences, we may probably group left 8 photos into one cluster and group the remaining photos into another cluster. Even more importantly, the textual tag could also be used to measure the “closeness” between the photos. Assume that there are four distinct tags that are “Art”, “Shopping”, “Food” and “Hotel” in this photo collection. The textual information

\(^6\)The longest piecewise subsequence can be viewed as a good local peak since it avoids to pick a value at head or tail levels. The value of head levels are fluctuated since the cost similarity in non-Gaussian distributed areas (i.e., big areas) is not stable. The value of tail levels is increasing since it requires large effort to decompose small areas.
of each photo is highlighted by different colors in the textual tag space. If we take all three spaces into consideration, the photos might be separated into three groups: the photos tagged by “Food” (in orange) and “Shopping” (in black) are grouped because they are highly relevant in both geographical and temporal spaces (probably the photos are taken around a shopping mall); the remaining photos are divided into two clusters: the ones tagged by “Hotel” (in green) and the ones about “Art” (in red).

![Fig. 8: The example of heterogenous information](image)

However, there is no explicit guideline to connect different aspects. One intuitive way is to combine them in a linear model that requires to have some weight settings to balance the contributions of these features. However, it is hard to define the proper weights for different datasets (cf. [Yang et al. 2011]). Furthermore, for the problem of POI identifications, the most fundamental feature is the geographical location of the photos since this feature explicitly provides the distribution of the photos in the spatial domain. Thereby, we treat the spatial feature as our main feature and the POI identification in spatial domain is thoroughly discussed in Section 3. Accordingly, we take the other features (e.g., temporal and textual) as the supportive features in this work.

These features can be used to further enhance the quality of the POI identification if we refine the self-tuning result of the spatial feature by the cohesion of these features. To be motivated by the previous discussion, we attempt to refine the clustering result produced by our self-tuning approach based on the supportive features. For such sake, we define a pair of metrics, **tightness** and **cohesion**, that assesses the inter-closeness of two clusters and the intra-closeness of a cluster, respectively. In this section, we carefully define the **tightness** and **cohesion** of two supportive features, i.e., temporal and textual features and study a framework to refine the self-tuning result by these two metrics.
4.1. Sequence-Based Tightness and Cohesion
In this section, we define the temporal cohesion of the clusters based on the time sequence of the photo collections. In this work, we capture the sequence relationship of the photos as the movement sequences from the same photographers. We denote the sequence relationship of photo (point) \( i \) and \( j \) as \( S = [S_{ij}]_{n \times n} \). Formally, we define the visit sequence relationship between two photos as follows.

\[
S_{ij} = \begin{cases} 
1 & \text{if } i \text{ and } j \text{ are taken by the same person consecutively} \\
0 & \text{otherwise}
\end{cases} 
\] (7)

![Diagram of tourist sequences]

Fig. 9: An example of tourist sequences

For instance, consider the example in Fig. 9, \( a_i, b_j, c_k, \) and \( d_l \) are taken by 4 different people, respectively. Therefore, we have \( S_{a_i, a_{i+1}} = 1, S_{b_j, b_{j+1}} = 1, S_{c_k, c_{k+1}} = 1, \) and \( S_{d_l, d_{l+1}} = 1. \)

Given the sequence relationships, we now define the sequence-based tightness between two clusters.

**Definition 4.1 (Sequence-Based Tightness between Two Clusters).** Given two cluster \( C_1 \) and \( C_2 \), the sequence-based tightness between \( C_1 \) and \( C_2 \) is defined as:

\[
tg_s(C_1, C_2) = \min\left\{ \sum_{i,j \in C_1} S(i,j), \sum_{i,j \in C_2} S(i,j) \right\}
\] (8)

where \( S(i,j) \) is the sequence relationship of \( i \) and \( j \).

It is clear that the sequence-based tightness of two clusters indicates the amount of travelers co-visit these two clusters. In other words, two clusters having high sequence-based tightness are more likely the same POI if they are spatially adjacent. With the sequence-based tightness, we can merge those clusters if they are overly partitioned by the self-tuning techniques.

**Definition 4.2 (Sequence-Based Cohesion of a Cluster).** Given a cluster \( C \), the sequence-based cohesion of \( C \) is defined as:

\[
coh_s(C) = tg_s(C_1, C_2)
\] (9)

where \( C_1 \) and \( C_2 \) are bi-partitioned from \( C \) using spectral clustering algorithm.

Vice versa, we can define the cohesion of a cluster (see Definition 4.2) based on the tightness function and its sub-clusters (collected from the execution tree of the spectral clustering), which indicates the internal closeness of a cluster. Similar to the tightness refinement, we can further partition a cluster \( C \) if the partitioning leads to a significant improvement of cohesion degree of \( C_1 \) and \( C_2 \) as compare to \( C \).
4.2. Text-Based Tightness and Cohesion of Clusters

In web photo albums, the textual tags are given by users when uploading their photos into the systems. These textual tags may provide additional information of the photos (e.g., the name of the landmark, specific event, or the name of surrounding area). However, as compared to spatial and temporal features, the textual information is less conformed since the tags are freely given by the users with no standard. According to our observation, we find the following properties of the photo textual information.

**Semantic Irrelevance.** For certain purposes, mobile devices may transparently insert some keywords into the metadata of the photos, such as “Nikon” and “Canon”, which are two most popular keywords found in the photo collections. Furthermore, some users prefer to plug their personal information into the metadata, such as the URL of their personal websites.

**POI Irrelevance.** According to our observation, a huge amount of tags are POI irrelevant. As summarized in [Overell et al. 2009], even with the assistance of Wikipedia and WordNet, the percentage of unrecognized textual tags is up to 30.8% and there are only 19.3% of the tags mentioned about their locations.

**Batch Processing.** The batch uploading tools significantly shorten the user uploading time; however, these tools may tag a set of location irrelevant photos in a batch. For instance, a sloppy tourist may tag all her photos by “Museum de Louvre” since she visited to “Museum de Louvre” in her trip.

Even though the textual information is quite different from the other features, these information still provide helpful information in identifying POIs. Taking Paris photo collection as an example, Fig. 10 illustrates that the photos tagged with “Museum” are densely distributed around the real museum locations. This inspires us to take the textual information into consideration by carefully selecting the tags.

---


---

ACM Transactions on Intelligent Systems and Technology, Vol. V, No. N, Article A, Publication date: January YYYY.
Table II: Textual Tags Photo Frequency (PF) for different cities

<table>
<thead>
<tr>
<th>City</th>
<th>No. Tags</th>
<th>No. Photos</th>
<th>Top 10 PF Textual Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paris</td>
<td>91291</td>
<td>796,427</td>
<td>Paris 708460, France 377237, Europe 72339, Louvre 42097, Francia 36492, Travel 34399, Nikon 29254, Eiffel 29055, Eiffel Tower 27626, Art 24845</td>
</tr>
<tr>
<td>New York</td>
<td>121033</td>
<td>1,087,964</td>
<td>New York 633324, Nyc 417340, New York City 337363, Manhattan 228543, New 172306, ny 171444, York 171003, USA 154115, Brooklyn 109259, city 107646</td>
</tr>
</tbody>
</table>

As shown in Table II, the tags are not equally important in terms of their frequency. In this work, we study several steps to filter the tags such that the less important tags are removed from the identification process.

1. Rule-based filtering: we define a set of manual rules that prunes pure numerical, date-related, and URL-based tags, such as “2009-05-11”, “www.flickr.com/photos/someone” and so on.
2. Photo Frequency (PF) filtering: as in Table II, the terms having highest PF are too general, such as country name and city name. Following the idea of Inverse Document Frequency (IDF), we remove those tags if they are widely used in many photos. In addition, we also remove those tags are less used in the photo collections in order to reduce the noisy. Accordingly, we use PF/IDF model to evaluate the significance of the tags.

Similar to the concept of sequence-based cohesion, now we define text-based tightness and cohesion of the clusters.

**Definition 4.3 (Text-Based Tightness between two Clusters).** Let $C_1$ and $C_2$ are two clusters, the text-based tightness between them is defined as:

$$tg(C_1, C_2) = \frac{C_1(t) \times C_2(t)}{||C_1(t)|| \times ||C_2(t)||}$$

(10)

where $C_1(t)$ and $C_2(t)$ are term vectors of the concatenated textual tags of photos in $C_1$ and $C_2$ respectively.

**Definition 4.4 (Text-Cohesion of a Cluster).** Given a cluster $C$, the text-based cohesion of $C$ is defined as:

$$coh_t(C) = tg(C_1, C_2)$$

(11)

where $C_1$ and $C_2$ are the two clusters bi-partitioned from $C$ using spectral clustering algorithm.

### 4.3. Refining Self-Tuning Clustering by Heterogenous Information

The self-tuning framework returns an execution tree (cf. Fig. 5(b)) based on the spectral clustering process. The leaf nodes of the tree $^8$ can be viewed as a set of good POIs in terms of their spatial information. However, they may be ineptly grouped or overly decomposed in terms of other features (cf. our discussion in Fig. 8). Thereby, it is interesting to see how to refine the self-tuning result by the tightness and cohesion of the supportive features.

$^8$The leaf nodes of the self-tuning are different from the spectral clustering due to the cut cost similarity assessment.
The refinement approach includes two phases: (1) partitioning and (2) merging, which are proposed to address those ineptly grouped and overly decomposed clusters, respectively. The partitioning phase takes the clusters produced by the self-tuning approach as input. It further partitions a cluster $C$ into $C_1$ and $C_2$ if the partitioning can lead to a significant improvement in terms of the cohesion degree. The partitioning operation recursively executes until no significant increase of the cohesion degree or the sizes of clusters become small enough. This phase attempts to improve the quality of those ineptly grouped clusters by the supportive features. More specifically, we define the significant increase of the cohesion degree as follows.

**Definition 4.5 (Significant Increase of Cohesion Degree).** Let $C_1, C_2, ..., C_N$ be clusters produced by the self-tuning spectral clustering. Then, we further partition a cluster $C_i$ into $C_{i,1}$ and $C_{i,2}$ if every feature $F$ in the feature set (e.g., $\{s,t\}$) satisfy

$$\frac{2\text{coh}_F(C_i)}{\text{coh}_F(C_{i,1}) + \text{coh}_F(C_{i,2})} < h^F_0$$

(12)

where $C_{i,1}$ and $C_{i,2}$ are the bi-partition spectral clustering result of $C_i$ and $h^F_0$ is a parameter.

Inequality 12 indicates whether the decomposition of $C_i$ offers much better cohesion degrees in terms of the supportive features. If it is satisfied, then we should replace $C_i$ by $C_{i,1}$ and $C_{i,2}$.

Accordingly, the merging phase takes the result of the partitioning phase as input. Similarly, it recursively merge two clusters into one unified cluster if the merging can lead to a significant improvement in terms of the tightness degree. Formally, the significant improvement of the tightness degree is defined as

**Definition 4.6 (Significant Increase of Tightness Degree).** Let $C_1, C_2, ..., C_N$ be clusters produced by the self-tuning spectral clustering. Then, we merge two spatially adjacent clusters $C_i$ and $C_j$ into $C_s$ if every feature $F$ in the feature set (e.g., $\{s,t\}$) satisfy

$$\frac{2\text{tg}_F(C_i, C_j)}{\text{coh}_F(C_i) + \text{coh}_F(C_j)} \geq h^F_0$$

(13)

Inequality 13 offers a merging condition to two spatially adjacent clusters. If there exist more than one pairs of spatially adjacent clusters satisfying Inequality 13, the one that maximizes $2\text{tg}_F(C_i, C_j)/(\text{coh}_F(C_i) + \text{coh}_F(C_j))$ is prioritized first. This procedure is executed recursively until no adjacent cluster satisfies the inequality.

The above refinement framework turns out a new issue: how can we set the value of $h^F_0$? Generally speaking, the cohesion (and tightness) degree of clusters increases along the downward partitioning since small clusters are probably more cohesive than big clusters. If we set $h^F_0 = h_{\text{max}}$, where $h_{\text{max}}$ is maximum cohesion degree among the clusters of the self-tuning result, the partitioning phase may overly partition the POIs. On the other hand, a small value of $h^F_0$ may lead to overly execute the merging operations. In this work, the clusters produced by our self-tuning algorithm are regarded as a good result in terms of spatial information. According to our discussion, a group of the self-tuning results are ineptly grouped while some of them are overly decomposed. Thereby, it makes sense to give a uniform assumption of these two cases as we do not know the distribution in advance. Accordingly, we simply take $h^F_{\text{med}}$, where $h^F_{\text{med}}$ is the median of cohesion among all self-tuning clusters based on the feature $F$ (see Equation 14), as the reference value for $h^F_0$ in the refinement algorithm. Our experimental
Actually, the partitioning and merging phases can be further reduced without much information loss. Where

\[ D(a, b) = \text{distance relationship} \]

the refinement process too much after normalization. One of the methods is to remove with a similarity in the strongest features and 2) the weak features do not involve into

There are two reasons for this: 1) photos are generally grouped together into the cluster of the photos across the groups. This is shown in Equation 15.

\[ X_{I,j}^G = \frac{\sum X_{ij}}{|X_{ij}|}, i \in I, j \in J \]  

where \( I \) and \( J \) are indicated two Hilbert curve groups of photos, respectively.

Furthermore, we should keep only substantial relations in the feature matrices. There are two reasons for this: 1) photos are generally grouped together into the cluster with a similarity in the strongest features and 2) the weak features do not involve into the refinement process too much after normalization. One of the methods is to remove a distance relationship \( D_{ij}^G \) from \( D^G \) if \( D_{ij}^G \) is larger than 200m. The computational cost can be further reduced without much information loss.

\[ \Omega(5) \text{ Return} \]

\[ \Omega = \{C_1, C_2, ..., C_N\} \]

\[ \forall F \in \{s, t\}, \text{median} (\text{coh}_F(C_i)) + \text{median} (\text{coh}_F(C_j)) \]  

\[ = \text{median}_{\Omega} \{2\text{coh}_F(C_i)/(\text{coh}_F(C_i) + \text{coh}_F(C_j))\}, \forall F \in \{s, t\} \]

\[ (14) \]

\[ h_m^F = \text{median}_{\Omega} \{2\text{coh}_F(C_i)/(\text{coh}_F(C_i) + \text{coh}_F(C_j))\} \]

**5. LARGE SCALE DATA HANDLING**

In this section, we discuss the implementation detail for our approach. Obviously, constructing a complete feature matrix for web scale datasets is too expensive. For instance, we have 216,300 geo-tagged photos in our Paris collection. For the distance matrix \( D \), it will consume approximate 183 GB space if all photo pairs are stored into \( D \). It is definitely too large for present computer systems.

To address the space issues, we first group the photos by Hilbert curve grouping [Lawder and King 2001] based on their distances, where the Hilbert curve grouping is a popular multi-dimensional indexing technique and well accepted by spatial databases. The grouping is shown to preserve very good quality to the original datasets, especially in low dimensionality. In this work, we carefully tune Hilbert curve parameters (e.g., maximum number of objects and grouping size) such that the photos collections are reduced to \( \sim 10\% \) of their original size.

After grouping, the group feature relationship \( X_{I,j}^G \) is set to the average feature relationship of the photos across the groups. This is shown in Equation 15.

\[ X_{I,j}^G = \frac{\sum X_{ij}}{|X_{ij}|}, i \in I, j \in J \]  

**Algorithm 2: Cluster Refinement**

**Input:** Clusters \( C_1, C_2, ..., C_N \), the results from Self-Tuning Clustering; Feature set \( F \in \{s, t\} \)

**Output:** Clusters \( C_1^G, C_2^G, ..., C_N^G \), the refined optimal clusters

1. \( h_0^F = \text{median}_{\Omega} \{2\text{coh}_F(C_i)/(\text{coh}_F(C_i) + \text{coh}_F(C_j))\}, \forall F \in \{s, t\} \)
2. \( \Omega = \{C_1, C_2, ..., C_N\} \)
3. For each \( C \in \Omega \), \( C_1 \) and \( C_2 \) are partitioned from \( C \), replace \( C \) with \( C_1 \) and \( C_2 \) if every feature \( F \in \{s, t\} \) satisfies \( 2\text{coh}_F(C)/(\text{coh}_F(C_1) + \text{coh}_F(C_2)) < h_m^F \)
4. For any adjacent pair \( C_1 \) and \( C_2 \) in \( \Omega \), replace \( C_1 \) and \( C_2 \) with \( C = C_1 \cup C_2 \) if every feature \( F \in \{s, t\} \) satisfies \( 2\text{tg}_F(C_1, C_2)/(\text{coh}_F(C_1) + \text{coh}_F(C_2)) \geq h_m^F \), \forall F \in \{s, t\} \)
5. Return \( \Omega \)

This reveals the effectiveness of our assumption. For clarity, we show the pseudo code of the refinement algorithm in Algorithm 2.
6. EXPERIMENTS
In this section, we present several experiments to demonstrate the superiority of our findings. We first introduce our data preparation and evaluation measures in Section 6.1 and Section 6.2, respectively. We compare our self-tuning spectral clustering to other clustering approaches in Section 6.3. At last, we investigate the effect of our refinement model in Section 6.4.

6.1. Data Preparation
We collect four photos collections (only meta data) with respect to four cities, Hong Kong, Paris, New York, and Rome, using Flickr API\(^\text{10}\). For each photo collection, we fetch all geo-tagged photos using the city name as the search key through the Flickr API. We filter out those photos that are not located in the city (by latitude and longitude boundary as shown in Table III) or have identical geo-location. The statistics of our photos collections are summarized in Table III.\(^\text{11}\)

<table>
<thead>
<tr>
<th>City</th>
<th>No. Photos</th>
<th>No. Clean Photos</th>
<th>Latitude and Longitude</th>
<th>No. MBRs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paris</td>
<td>796,427</td>
<td>216,300</td>
<td>48.8151 48.9030 2.2232 2.4742</td>
<td>297</td>
</tr>
<tr>
<td>Hong Kong, HK</td>
<td>193,369</td>
<td>43,294</td>
<td>22.17596 22.56773 113.70913 114.44252</td>
<td>28</td>
</tr>
<tr>
<td>New York, NY</td>
<td>1,067,964</td>
<td>274,428</td>
<td>40.49866 40.93115 41.79 41.989</td>
<td>97</td>
</tr>
<tr>
<td>Rome</td>
<td>343917</td>
<td>96738</td>
<td>41.79 41.989 12.368 12.624</td>
<td>102</td>
</tr>
</tbody>
</table>

6.2. Evaluation Measure
To evaluate the performance of our proposed approaches, we extract a set of tourist attractions in the metadata of OpenStreetMap as our ground truth for each city. The total number of attractions for each city is shown in Table III. The tourist attractions are represented by MBRs as shown in Fig. 2. In the evaluation, the MBRs are enlarged to 150% of their original size since photos might be taken in the surrounding area as well. An POI is perfectly identified by a cluster if and only if all photos in the MBR are grouped into the cluster and their sizes are identical. In the evaluation, we filter out the clusters which do not intersect to any MBR.

In IR community, a common way to interpretation of clustering is to view it as a series of decisions, one for each of the \(N(N - 1)/2\) pairs of photos in the collection. We classify these decisions by a simple binary classification. For instance, a true positive decision assigns two photos in the same MBR to the same cluster C too. \(TP\) is used to indicate the total number of true positive decisions. We summarize other types of decisions in Table IV.

<table>
<thead>
<tr>
<th>Type</th>
<th>(P(Positive))</th>
<th>(N(Negative))</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP(\text{True})</td>
<td>C, MBR</td>
<td>¬C, ¬MBR</td>
</tr>
<tr>
<td>FP(\text{False})</td>
<td>C, ¬MBR</td>
<td>¬C, MBR</td>
</tr>
</tbody>
</table>

\(^{10}\)http://www.flickr.com/services/api/
\(^{11}\)The counted Tags are referenced by two or more photos
In [Manning et al. 2008], the precision $P$, recall $R$, and $F_\beta$ measure are defined by Equation 16. In this paper, we use the $F_1$ measure as our evaluation measure. A large value of $F_1$ measure indicates a better clustering.

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$ (16)

6.3. Self-tuning Evaluation

<table>
<thead>
<tr>
<th>Clustering</th>
<th>Default Settings</th>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$-means</td>
<td></td>
<td>$k$</td>
<td>20, 50, 100, 200, 500, 1000, 2000</td>
</tr>
<tr>
<td>fuzzy $c$-means</td>
<td>$m = 2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$g$-means</td>
<td></td>
<td>$c$</td>
<td>20, 50, 100, 200, 500, 1000, 2000</td>
</tr>
<tr>
<td>mean shift</td>
<td>Gaussian kernel, $\sigma^2 = 0.2$</td>
<td>$\eta$ (in meter)</td>
<td>10, 25, 50, 100, 200, 400</td>
</tr>
<tr>
<td>P-DBSCAN</td>
<td>$\theta = 50, \omega = 0.1$</td>
<td>$\varepsilon$ (in meter)</td>
<td>10, 25, 50, 100, 200, 400</td>
</tr>
<tr>
<td>spectral clustering</td>
<td></td>
<td>$\delta$</td>
<td>0, 0.05, 0.1, ..., 0.9, 0.95, 1</td>
</tr>
</tbody>
</table>

In this sub-section, we demonstrate the superiority of our self-tuning technique using our photos collections. Fig. 11 shows the effect of the subjective parameter of different clustering approaches. Besides the approaches introduced in Section 2, we additionally compare two variants of $k$-means, fuzzy $c$-means [Nock and Nielsen 2006] and $g$-means [Hamerly and Elkan 2003]. The fuzzy $c$-means clustering attempts to enhance the clustering performance by fuzzy techniques and the $g$-means clustering can automatically determine the number of clusters for Gaussian distributed datasets. For each evaluated clustering approach, we only vary the main parameter while setting the others to their default values as listed in Table V. As a remark, we carefully tune the best gaussian threshold for the $g$-means clustering for fairness.

We first compare our self-tuning technique to these clustering approaches using the Paris photo collection. According to results in Fig. 11, it is obvious that our self-tuning technique outperforms the others. Without any parameters tuning, its $F_1$ measure is only worse than one case in mean shift ($\eta = 100m$), one case in P-DBSCAN ($\varepsilon = 50m$). Our self-tuning reports similar performance to the best results of the spectral clustering (e.g., $\delta = 0.1$ and $\delta = 0.45$). It is because our approach can self-identify the dense region by the cut costs similarity instead of using a global threshold $\delta$. The experiment results on New York photos are very similar; as shown in Fig. 12, our self-tuning is slightly worse than the optimal cases of the spectral clustering and P-DBSCAN clustering, and surprisingly, it beats Mean Shift Clustering among all evaluated parameters.

Note that our self-tuning approach might be worse than other clustering approaches when we know their best parameter settings in advance. However, as demonstrated by Fig. 11 and Fig. 12, the self-tuning approach provides an acceptable quality without manually tuning. As a reference, we illustrate the snapshots of the self-tuning approach in Fig. 13. We believe that our self-tuning technique is not only valuable to the clustering problem (i.e., POIs identification) but also to other trip related mining problems.

Fig. 14 compares the $F_1$ measures between the spectral clustering and our self-tuning approach using the Hong Kong and Rome photo collections. In Hong Kong, again our self-tuning technique outperforms the spectral clustering for all tested $\delta$ values. The $F_1$ measure of Hong Kong collection is small since the tourist attractions provided by OpenStreetMap are very limited and the region of the attractions is very small. For the Rome collection, the result of the self-tuning again outperforms the spectral clustering in most settings. As mentioned previously, we aim to find the good result
Identifying Points of Interest using Heterogenous Features

Fig. 11: $F_1$ measures for different clustering on Paris photo collection

without manual tuning. The result of these two photo collections (Hong Kong and Rome) demonstrates that our technology can well handle this task.

6.4. Clustering Refinement using Heterogenous Features

In this sub-section, we study the effect of the clustering refinement discussed in Section 4 using two representative photo collections, Paris and New York. We ignore other photo collections since they do not provide enough heterogenous information for the refinement process.

Table VI: The effectiveness of refinement on different features

<table>
<thead>
<tr>
<th>City</th>
<th>Self</th>
<th>Self + Textual, $\mathcal{F} \in {t}$</th>
<th>Self + Temporal, $\mathcal{F} \in {s}$</th>
<th>Self + BOTH, $\mathcal{F} \in {s,t}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paris</td>
<td>0.459</td>
<td>0.487</td>
<td>0.605</td>
<td>0.611</td>
</tr>
<tr>
<td>New York</td>
<td>0.479</td>
<td>0.483</td>
<td>0.497</td>
<td>0.570</td>
</tr>
</tbody>
</table>

We compare our self-tuning result to three different refinement approaches that take different set of supportive features into consideration, which are shown in Table VI. For instance, Self + Temporal indicates a method refines the self-tuning result only by the temporal feature, i.e., $\mathcal{F} \in \{s\}$, (cf. Section 4.1). It is clear that our refinement technique significantly improve the $F_1$ score over the self-tuning technique. If we take all supportive features into the refinement approach, the improvement over the self-tuning is quite significant which is 31.8% and 19% on Paris and New York photo collections, respectively. This shows that our refinement techniques are useful for POI identification.
We illustrate the clustering result after the refinement process in Fig.15. As compared to the illustration of the self-tuning technique (see Fig.13), the POI “Musée du Louvre” in Paris and “Rockefeller Center” in New York (both are highlighted by purple color) are identified more precisely.

We evaluate the effect of the parameter $h_{0}^{F}$. We vary the value of $h_{0}^{F}$ from $h_{\text{min}}$ to $h_{\text{max}}$, where $h_{\text{min}}$ indicates the minimum cohesion degree among the clusters of the
Identifying Points of Interest using Heterogenous Features

Fig. 14: Comparison between spectral clustering and self-tuning on Hong Kong and Rome photo collections.

Fig. 15: Refinement illustration on Paris and New York photo collections.

Fig. 16: “Reinforced” Clustering Results On Paris and New York Photo Collections By Varying $h^{F}_0$

self-tuning result. The F1 score of different $h^{F}_0$ settings are shown in Fig.16. These two figures show that our assumption (i.e., setting $h^{F}_0$ to the medium value $h_m$) is quite
reasonable as the value of $h_0^F$ that maximizes F1 is not far from $h_m$. In addition, the performance is quite stable when the value of $h_0^F$ is more or less the same of $h_m$.

6.5. Computation Time

<table>
<thead>
<tr>
<th>Approach</th>
<th>Paris</th>
<th>New York</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$-means</td>
<td>20.21s</td>
<td>22.35s</td>
</tr>
<tr>
<td>$f$-means</td>
<td>120.49s</td>
<td>147.81s</td>
</tr>
<tr>
<td>Fuzzy $c$-means</td>
<td>425.54s</td>
<td>445.12s</td>
</tr>
<tr>
<td>P-DBSCAN</td>
<td>24.81s</td>
<td>25.76s</td>
</tr>
<tr>
<td>Mean shift</td>
<td>311.97s</td>
<td>412.43s</td>
</tr>
<tr>
<td>Spectral Clustering</td>
<td>1015.63s</td>
<td>563.43s</td>
</tr>
<tr>
<td>Self-tuning (Section 3.2)</td>
<td>1018.80s</td>
<td>567.40s</td>
</tr>
<tr>
<td>Refinement (Section 4.3)</td>
<td>1094.88s</td>
<td>646.72s</td>
</tr>
</tbody>
</table>

In this section, we demonstrate the execution time of our evaluated approaches. For the ease of presentation, we only report the performance of the approaches by setting all parameters to their default values (see Table V). All methods were implemented in Java and Matlab runtime 2011b (for Eigenvalue computation only). The results were conducted on an Intel I7 830 2.93GHz CPU machine with 8 GBytes memory, running on Win7 64bit. Table VII shows the execution time of different approaches on two datasets, Paris and New York. Among all prior approaches, spectral clustering has the slowest execution time due to a sequence of expansive cut processes. To make things worse, our implementation uses an external library (i.e., Matlab runtime) to compute the eigenvalue which move abundant data in-between the main Java program and the Matlab runtime. Obviously, both of our methods (i.e., self-tuning and refinement) are based on spectral clustering approach. Their execution overheads are negligible as shown in Table VII. This indicates that our proposed techniques are not only effective but also efficient. As a remark, the overall performance of our approaches can be significantly boosted by the latest development of the spectral clustering [Hein and Bühler 2010; Szlam and Bresson 2010; Hein and Setzer 2011]. As the execution time is not our main focus in this work, we leave this issue in our future work.

7. RELATED WORK

There are many mining work based on web photo collections. Rattenbury et al. [Rattenbury et al. 2007] discover the names of events and landmarks using geo-tagged and textual data of photos from Flickr. Kennedy and Naaman [Kennedy and Naaman 2008] and Crandall et al. [Crandall et al. 2009] study a problem which identify landmarks by different clustering algorithms based on geo-tagged photos. The result in [Crandall et al. 2009] inspires many subsequent work. Their work also propose a classification approach to identify the location of a photo based on both visual and textual features. However, these works do not take temporal information into the identification process.

Besides POI identification, there are many recent work to deduce trip information from web photos. Clements et al. [Clements et al. 2010] study a work that predicts travel interest of a user who had rich travel data in the past. Popescu and Grefenstette [Popescu and Grefenstette 2009] attempt to deduce visit duration in tourist attraction. The authors first clean the photo collection by a set of heuristic filtering. Then, they

\[\text{We do not find any native Java library to support Eigenvalue computation in large sparse matrices.}\]
apply an external coverage geographical gazetteer to map the photos to different POIs based on their textual and geo-tagged information. Their results are subsequently used in different trip-related tasks, such as maximum, minimum, and average stay time of POIs. In [Popescu et al. 2009], the authors study to extract day-tour information, such as people visit interests, visit time, and duration time. Their solution first identifies POIs by an external knowledge base, e.g., Wikipedia, and then extracts the day-tour information based on the identified POIs. Based on the same notion that external knowledge resources provide precise landmark information, Popescu and Grefenstette in [Popescu and Grefenstette 2011] design a system to suggest tourist personalized visit recommendations based on the landmark description in Wikipedia.

Automatic tour planning from geo-tagged photos has been studied in [Choudhury et al. 2010; Jain et al. 2010]. Choudhury et al. [Choudhury et al. 2010] constructs intra-city travel itineraries using spatio-temporal data from Flickr. Their solution first identifies a set of time paths, where each time path is a sequence of POIs traversed by a user. The duration time and the transit time of POIs are subsequently extracted from the time paths. Antourage [Jain et al. 2010] automatically constructs tourist trip from the geo-tagged photos, by specifying a start location and the maximum distance of a trip. In their work, the map is formed by a hexagonal overlay grid, where each hexagonal cell is weighted by the number of photos taken inside. The tour planning problem in their work becomes an optimization problem that maximizes the total weight of the selected cells where the total distance fulfills the given constraint.

User travel pattern mining is another popular research topic in recent years. Zheng et al. [Zheng et al. 2012] build a complete framework that aims to detect the travel patterns of user trajectories using three types of features: geographical, temporal and textual, where the POI identification in their work is done by DBSCAN. Another active research topic is to produce diverse or personalized travel recommendation to users. In [Kurashima et al. 2010], Kurashima et al. design a system that recommends the travels by estimating the probabilities of a user visiting landmarks, where the landmarks are identified based on the mean shift clustering. Cheng et al. [Cheng et al. 2011] demonstrate the effectiveness of interpolating the user attributes(gender, age and so on) during the tourist recommendation process.

According to Observation 1, the photos in POIs are more likely Gaussian distributed. There are several work developed based on similar observation. Cooper [Cooper 2011] proposes a large scale photo clustering framework which aims to generate the photo clusters having cohesion in geographical and temporal space, where their spatial cohesion is based on a Gaussian kernel. Huang et al. in [Huang et al. 2012] attempt to detect the POI viewing angle by user query logs, where their detection is also based on a Gaussian kernel as well.

Regarding the parameter-free techniques in clustering, there are some result reported in machine learning community. In [Pelleg and Moore 2000], Pelleg et al. design an approach that aims to find the optimal value of $k$ in the $k$-means clustering based on a set of statistical tests. Zelnik-Manor et.al in [Zelnik-Manor and Perona 2004] propose a self-tuning spectral clustering that automatically determines the optimal cluster number using eigenvector analysis. Given a normalized affinity matrix, the algorithm attempts to minimize the cost of aligning its top $\Delta$ eigenvectors with a canonical coordinate system, where $\Delta$ indicates the largest possible cluster numbers. It incrementally evaluates the alignment cost of every cluster numbers (from 2 up to $\Delta$). However, this approach is not scalable as $\Delta$ may become very large in web scale photo collections. We experimentally evaluate [Zelnik-Manor and Perona 2004] and their result prefers a small number as the optimal number of clusters in our photo collections. Tibshirani et al. [Tibshirani et al. 2000] propose a self-tuning framework based on a Gap statistic testing. Their work compares the likelihood of the result to the training...
data. Again, their algorithm is not scalable to the number of clusters according to their experimental study. Feng et al. [Feng and Hamerly 2006] propose the pg-means clustering which assumes that the data are generated from a Gaussian mixture. However, this clustering approach requires to set a global parameter for Kolmogorov-Smirnov (KS) test, which is not a completely parameter-free approach. Hamerly et al. [Hamerly and Elkan 2003] propose the g-means algorithm which iteratively bi-partitions a cluster until all sub-clusters pass a Gaussian distribution testing. However the Gaussian distribution testing requires to set a threshold which is not applicable to POI identification as discussed in Section 1. In summary, these algorithms may well work for other applications but not for the POI identification due to the scalability of the data.

8. CONCLUSIONS

In this paper, we analyze the POI identification problem by thoroughly comparing the state of the art approaches and figuring out two issues: (1) parameter is subjective and sensitive in different clustering algorithms and (2) existing clustering approaches highly depends on the geographical information of the photos which are not enough for identifying high quality POIs. Regarding the first issue, we study spectral clustering which is the first attempt for POIs identification. In addition, according to spectral clustering procedures, we propose a self-tuning approach based on a novel cut cost similarity such that the clustering process becomes parameter-free. To address the second issue, we study a novel refinement model that is based on the concept of tightness and cohesion degree of the clusters in different features. We thoroughly evaluate our proposed algorithms in our experimental section using four Flickr photo collections. Our proposed algorithms provide excellent results while they are all parameter-free.

In the future, we are intent to study the POIs ranking problems in terms of attractiveness. In general, the POIs can be ranked by simple observation such as number of photos, number of users, or size. It is definitely interesting to investigate a unified approach to rank properly by these factors. In addition, identifying POIs can be viewed as a module of other trip related mining tasks such as the theme of POIs. We will investigate how these related tasks can be improved by our clustering framework.

REFERENCES

Identifying Points of Interest using Heterogenous Features

HAMERLY, G. AND ELKAN, C. 2003. Learning the k in k-means. In NIPS.