

Online Clustering Algorithms for Semantic-Rich Network Trajectories

Gook-Pil Roh and Seung-won Hwang*

Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Korea
noh9pil@postech.ac.kr, swhwang@postech.ac.kr

Abstract

With the advent of ubiquitous computing, a massive amount of trajectory data has been published and shared in many websites. This type of computing also provides motivation for online mining of trajectory data, to fit user-specific preferences or context (e.g., time of the day). While many trajectory clustering algorithms have been proposed, they have typically focused on offline mining and do not consider the restrictions of the underlying road network and selection conditions representing user contexts. In clear contrast, we study an efficient clustering algorithm for Boolean + Clustering queries using a pre-materialized and summarized data structure. Our experimental results demonstrate the efficiency and effectiveness of our proposed method using real-life trajectory data.

Category: Ubiquitous computing

Keywords: Online clustering; Semantic-rich trajectory

I. INTRODUCTION

With the advent of ubiquitous computing devices equipped with global positioning system (GPS), radio frequency identification (RFID) or sensors, we can easily record object movements or trajectories. Recently, much research work [1-4] has aimed to evaluate spatio-temporal queries, such as range and nearest neighbor queries, on a trajectory database from the given query point.

While most existing work focuses on supporting trajectories that record only the locations over time, the actual trajectories collected from recent location-based services often have greater semantic richness. For example, Flickr (flickr.com) allows users to geo-tag their photos and annotate them with comments and tags. This enables users to describe their trips as a trajectory of photos and information, for example, Fig. 1 shows a geo-tagged photo of the opera house on the map with annotated information such as title, username or timestamp. Such additional information enables us to bridge the semantic gap between the location and the actual activity at the location, by distinguishing between

different activities that occurred at the same location, or the same activities at different locations, as Fig. 2 illustrates. Recent work [5], distinguishing activities from locations using user comments, has the same goal as ours: supporting semantic-rich location-based service.

The scale of such semantic-rich trajectories, e.g., millions of geo-tagged images added monthly to Flickr, naturally motivates us to support advanced data analysis (e.g., clustering), e.g., recommending interesting travel itineraries [6].

Our goal is to enable online clustering of semantic-rich trajectories, motivated by the following example scenario.

Example 1. While existing forms of travel recommendation, e.g., travel guides or portals, focus on offline computation rendering the same itinerary for all users, travel recommendation on ubiquitous devices naturally motivate forms of online recommendation that can adapt to changes in user-specific context. For instance, the recommendation should vary according to the temporal context, such as time of the day or season of the year.

We thus propose to study online trajectory clustering algorithms, combining the following semantic context criteria:

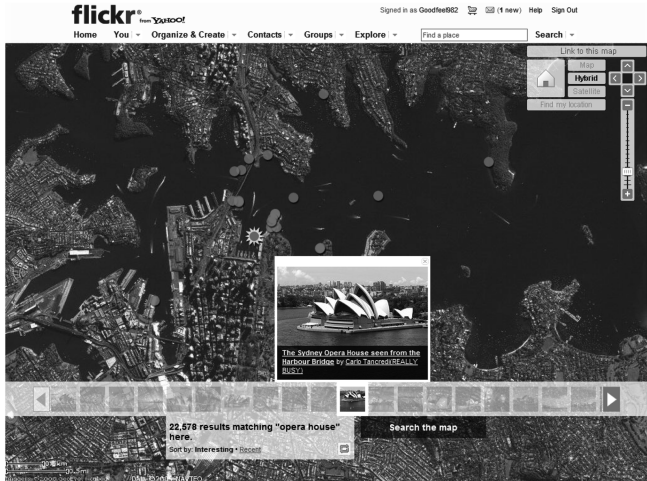
Open Access <http://dx.doi.org/10.5626/JCSE.2011.5.4.346>

<http://jcse.kiise.org>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

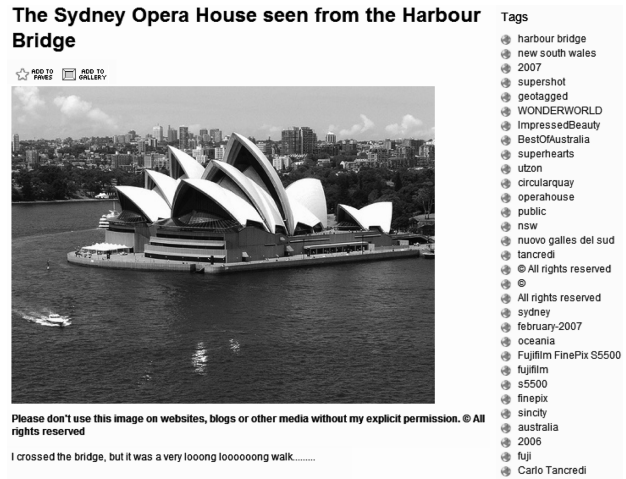
Received 01 July 2011, Accepted 26 August 2011

*Corresponding Author



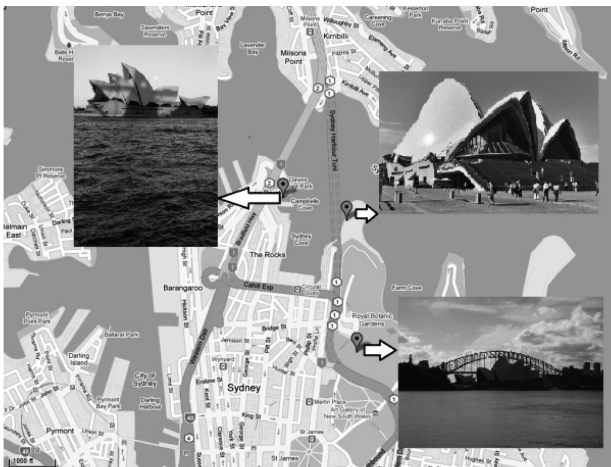
(a) Search result on Flickr

{tag="opera house"; location="Sydney"}

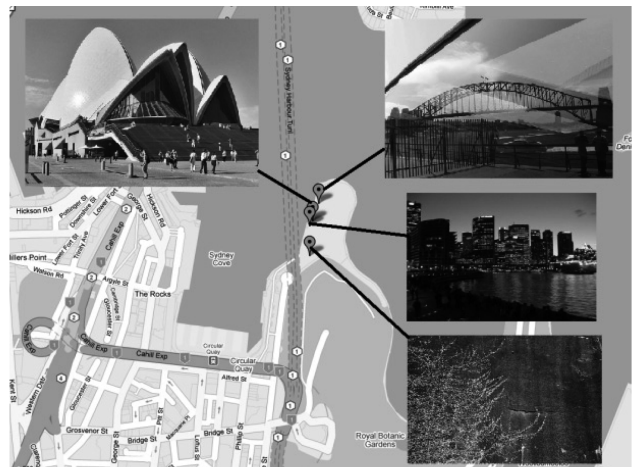


(b) Image and tags

Fig. 1. Geo-tagged image example on Flickr.



(a) Same activity over different locations



(b) Diverse activities at a single location

Fig. 2. Semantic gaps between location and activity.

- Soft criteria: Users may embed various semantic information, e.g., semantic geographic information, in the similarity measure.
- Hard criteria: Users may consider only trajectories satisfying some Boolean semantic criteria, such as time = evening or user = Tom.

In this paper, we focus on combining the two semantic context criteria with clustering, which can be easily provided by the user at the query time. However, existing clustering algorithms cannot address this problem due to the following challenges. First, most existing work focuses on offline unconstrained location trajectories, such as hurricanes or animal trajectories. In contrast, in many location-based services, trajectory mining should reflect the proximity with respect to the underlying constraint, e.g., road networks or subway routes, because two nearby locations can be very hard to reach if no roads connect

the two locations. We study online clustering, which supports soft semantic criteria using a similarity measure based on the underlying geographic information such as road networks.

Secondly, to support hard semantic criteria, we need to support Boolean + Clustering queries efficiently. While existing clustering algorithms can be adopted after Boolean conditions are evaluated, such naive adoption incurs a significantly high computational cost, which provides motivation for an integrated algorithm capable of evaluating both.

To this end, we design a summarized data structure where pre-computed initial clusters of trajectories are stored. This can expedite online computation in order to merge these initial clusters. We store statistical information on selection attributes to reflect Boolean selections in the merging process.

This paper addresses the above challenges, which is an extension of our preliminary work [7] that only tackled the first

challenge. We discuss our extension of this preliminary work in Section II. The main contributions of this paper are as follows:

- We are motivated to study the problem of supporting Boolean + Clustering queries for semantic-rich trajectories.
- We design a summarized structure for pre-materializing initial trajectory clusters.
- We propose an efficient clustering algorithm built on the proposed summarized structure.
- We extensively validate the effectiveness and the efficiency of our proposed framework using real-life trajectory data.

The rest of the paper is organized as follows. Section II presents the prior research work. In Section III, we present the formal definition of our problem. Section IV studies distance measures. Section V describes our clustering algorithms. Section VI reports our experimental results. Section VII concludes this paper.

II. RELATED WORK

Lately, clustering trajectories have been studied actively, but most work assumes trajectories without road network constraints, as in hurricane tracking or animal movement data. For trajectories without constraints, trajectories are typically represented by either a set or sequence of locations (sampling points), for which similarity metrics have been proposed in [8-13] and [14-17], respectively.

In a clear contrast, this work focuses on trajectories that are constrained by road networks, for which a similarity metric has only recently been proposed in [7]. In many real-life applications, moving objects are often constrained by obstacles such as buildings, trees, or roads. Roh and Hwang [7] studied a similarity metric that reflects such constraints in trajectory analysis.

Once similarity metrics are defined, existing clustering algorithms can be applied to group trajectories into clusters. Specifically, recent work [18] proposes to adopt the expectation maximization (EM) algorithm and density-based clustering algorithm [12] for the task of clustering trajectories. However, these algorithms are not suitable for our proposed problem of online clustering with Boolean selection conditions. Gaffney and Smyth [18] requires slow convergence, which prohibits online responses, while [12] requires to pre-materialize an index structure for online computation, which will be ineffective when there exist Boolean selection conditions.

Our preliminary work [7] distinguishes itself from the existing work by designing a distance function for network-constrained location trajectories and developing an online algorithm. Based on this preliminary work, we extend our algorithms to support semantic-rich trajectories, archiving not only location changes, but also semantic information such as images, text, user ID and tags.

In particular, our study aims to support Boolean selection of semantic features in this paper. For efficient support, we propose to minimize online computational requirements, as similarly adopted for numerical data in [19]. However, adopting [19] for our proposed problem is not desirable, as this work assumes two objects with similar selection attribute values are also deemed to be similar according to the similarity metric. This assumption does not hold in our problem, where similarity

and selection attributes are allowed to be independent, e.g., similarity of image features and selection of timestamps.

The most relevant work [5] to this paper distinguishes activities from locations using comments added by users on the trajectory. This work has the same goals as ours; we support Boolean conditions or texts or tags, such as $tag = dining$. A key distinction of our work from the existing work is to provide a systematic framework for online clustering with arbitrary Boolean semantic conditions.

III. PROBLEM DEFINITION: BOOLEAN + CLUSTERING

In this section, we formally define our problem. We begin by modeling a network-constrained trajectory. We model a road network as a graph, where an edge is a road segment and a vertex is an intersection of adjacent segments, formally stated as follows:

Definition 3.1 Road network. A road network (RN) is defined as a graph $G_{RN} = (V, E)$, where V is a set of intersections of the road network, and E is a set of road segments $R_i \in E$ such that

$$R_i = (r_{i,s}, r_{i,e}),$$

where $r_{i,s}, r_{i,e} \in V$ and there exists a road between $r_{i,s}$ and $r_{i,e}$.

The trajectories on the road network can thus be represented by a set of connected road segments, or edges in the above-mentioned graph. Methods of matching GPS logs to road segments have been studied in existing map matching literatures (e.g., [20]).

Definition 3.2 Network-constrained trajectory. Given a road network $G_{RN} = (V, E)$, a trajectory of length l is defined as

$$TR_i = [t_1^i, \dots, t_l^i],$$

where $t_j^i \in E, 1 \leq j \leq l$ and t_j^i and t_{j+1}^i are connected.

We formally define our problem of finding a set of disjoint clusters C for trajectories that satisfy the given Boolean condition set B . The Boolean condition set can be a complex Boolean condition such as conjunctions and disjunctions of sub-conditions.

Definition 3.3 Problem statement. Given input trajectories T that satisfy Boolean selection condition set B , our goal is to find $C = \{c_1, \dots, c_{|C|}\}$, maximizing the sum of inter-cluster distances, such that

- (1) $c_i \subset T$ and $c_i \neq \emptyset, 1 \leq i \leq |C|$;
- (2) $\bigcup_{i=1}^{|C|} c_i = T$;
- (3) $c_i \cap c_j = \emptyset, 1 \leq i, j \leq |C|$ and $i \neq j$;

where the inter-cluster distance D_{inter} is defined by the distance between representative trajectories of two clusters. Note that our target problem does not require users to specify a fixed number of clusters.

Definition 3.4 Inter-cluster distance. Given two clusters, c_i and c_j , the distance between them is defined by

$$D_{inter}(c_i, c_j) = D(rt(c_i), rt(c_j)), \quad (1)$$

where $rt(c_i)$ is the representative trajectory of a cluster c_i , i.e.,

$$rt(c) = \underset{TR_i \in c, m}{\operatorname{argmin}} \frac{1}{m} \sum_{j=1}^m D(TR_i, TR_j) \quad (2)$$

and $D(\cdot, \cdot)$ is the trajectory distance, which we will define in the next section.

IV. DISTANCE MEASURE

In this section, we discuss a distance measure between network-constrained trajectories. To define a desirable distance measure, we have to consider the following two requirements:

Accuracy: The distance measure should accurately reflect the spatial proximity from the viewpoint of the road network.

Efficiency: The distance measure should be a metric for efficient computation of clustering, as we will elaborate in the details in Section V.

While distance measures satisfying the above requirements for trajectories without road network constraints have been actively studied [12-15], the distance measure for network-constrained trajectories has only recently been proposed in our preliminary work.

A. Accuracy Requirement

To address the first requirement that spatial proximity must be accurately represented, our distance measure adopts the intuitive notion of the Hausdorff distance to quantify the distance of two trajectories TR_i and TR_j as the longest road network path from each road segment to its closest road segment in the other trajectory. More formally,

Definition 4.1 Trajectory distance [7]. Given two trajectories $TR_i = [t_1^i, \dots, t_n^i]$ and $TR_j = [t_1^j, \dots, t_m^j]$, the distance between them is defined as follows:

$$D(TR_i, TR_j) = \max\{\vec{D}(TR_i, TR_j), \vec{D}(TR_j, TR_i)\}, \quad (3)$$

where $\vec{D}(TR_i, TR_j)$ is the one-way trajectory distance from TR_i to TR_j , defined as:

$$\vec{D}(TR_i, TR_j) = \max_{t_a^i \in TR_i} \min_{t_b^j \in TR_j} d(t_a^i, t_b^j). \quad (4)$$

Meanwhile, the road segment distance $d(\cdot, \cdot)$ in the previous definition is defined as follows:

Definition 4.2 Road segment distance [7].

$$d(R_p, R_j) = \max\{\vec{d}(R_p, R_j), \vec{d}(R_j, R_p)\}, \quad (5)$$

where $\vec{d}(R_p, R_j)$ is a one-way road segment distance from R_i to R_j defined as:

$$\vec{d}(R_p, R_j) = \max\left\{ \min\{\|r_{i,s}, r_{j,s}\|, \|r_{i,s}, r_{j,e}\|\}, \min\{\|r_{i,e}, r_{j,s}\|, \|r_{i,e}, r_{j,e}\|\} \right\}, \quad (6)$$

where $\|a, b\|$ is the length of the shortest path between a and b .

B. Efficiency Requirement

To address the second requirement of efficient computation, we show that our distance measure is a metric.

Theorem 4.3. Trajectory distance in Definition 4.1 is a metric.

Proof. Given three trajectories TR_i , TR_j , and TR_k , Equation 3 and 4 satisfy the following properties.

$$(1) \vec{D}(TR_i, TR_j) = 0 \text{ iff } TR_i \subset TR_j$$

$$(2) \vec{D}(TR_j, TR_i) = 0 \text{ iff } TR_j \subset TR_i$$

$$(3) D(TR_i, TR_j) = D(TR_j, TR_i)$$

$$(4) D(TR_i, TR_k) \leq D(TR_i, TR_j) + D(TR_j, TR_k)$$

The symmetric property follows from 3, the reflexive property follows from 1 and 2, and the triangle in equality follows from 4.

As the symmetric and reflexive properties are obvious from the definition, we only present the proof of triangle in equality. First we consider ε -neighborhood road segments of a trajectory, denoted by $N_\varepsilon(TR)$, which are road segments within ε distance from any road segment in TR .

$$N_\varepsilon(TR) = \{R \in RN \mid d(R, t_m^i) \leq \varepsilon, 1 \leq m \leq n\}.$$

With this notion, the proof of triangle inequality is as follows. For convenience, we shortly denote $\vec{D}(TR_i, TR_j)$ by \vec{D}_{ij} .

$$\begin{aligned} \vec{D}_{ik} &= \inf\{\varepsilon > 0 \mid TR_i \subset N_\varepsilon(TR_k)\} \\ &= \inf\{\delta + \delta' \mid TR_i \subset N_{\delta+\delta'}(TR_k)\} \\ &\leq \inf\{\delta + \delta' \mid TR_i \subset N_\delta(TR_j) \text{ and } TR_j \subset N_{\delta'}(TR_k)\} \\ &\leq \inf\{\delta \mid TR_i \subset N_\delta(TR_j)\} + \inf\{\delta' \mid TR_j \subset N_{\delta'}(TR_k)\} \\ &= \vec{D}_{ij} + \vec{D}_{jk} \end{aligned} \quad (7)$$

Similarly,

$$\vec{D}_{ki} = \vec{D}_{kj} + \vec{D}_{ji} \quad (8)$$

From Equation 7 and 8, the following inequality holds.

$$\max\{\vec{D}_{ik}, \vec{D}_{ki}\} \leq \max\{\vec{D}_{ij}, \vec{D}_{ji}\} + \max\{\vec{D}_{jk}, \vec{D}_{kj}\} \quad (9)$$

$$\therefore \vec{D}(TR_i, TR_k) \leq \vec{D}(TR_i, TR_j) + \vec{D}(TR_j, TR_k)$$

V. SEMCLUSTER ALGORITHM

In this section, we present a clustering algorithm called SemCluster, which supports Boolean + Clustering queries for semantic-rich network trajectories. As NNCluster [7] is an online algorithm, it can be naively adopted for Boolean + Clustering queries, by evaluating Boolean selection first and performing NNCluster only on the qualifying results. However, such naive adoption incurs a high cost, especially when the selectivity of Boolean conditions is high.

To address this challenge, the basic concept of SemCluster involves minimizing the distance computations at runtime by pre-materializing initial clusters into a summarized data structure. Based on this concept, we devise SemCluster with two key components—BuildDS and MergeDS. The BuildDS module materializes the summarized data structure from trajectories in an offline process. The MergeDS module performs Boolean + Clustering queries with the pre-materialized summarized data structure.

As we exploit the same property, sharing common nearest

neighbors, of NNCluster when we identify initial clusters, we first give a brief review on identifying initial clusters in Section V-A. We then present the BuildDS module in Section V-B and the MergeDS module in Section V-C.

A. Review: Identifying Initial Clusters

We briefly review our method of identifying initial clusters of trajectories. We group the trajectories into initial clusters that are likely to belong to the same cluster. To decide which trajectories to put into the same initial cluster, we need to identify a set of trajectories sharing the common nearest neighbor. This problem can be abstracted as a graph problem, by denoting each trajectory as a node connected to another node that is the nearest neighbor. Given this graph, the problem of building the initial clusters can be reduced to listing all maximal cliques, which is known as an NP-complete problem [21].

Our goal is to approximate this process by relaxing the notion of initial clusters to those sharing many *k*-nearest neighbors. To effectively construct initial clusters, we compare the *k*-nearest neighbors of each trajectory with those of another trajectory. The two trajectories are then merged into a cluster if the ratio between the common nearest neighbors is greater than a user-defined threshold value. Interested readers may find the details on identifying initial clusters and setting the threshold value in [7].

B. BuildDS

This section describes our method of pre-materializing a summarized data structure from the trajectories in the database. The summarized data structure consists of two components: 1) initial cluster information and 2) the distance relationship between initial clusters.

Once initial clusters are identified, we store the information of the initial clusters, such as the representative trajectory of the initial cluster and the number of members. We compute the representative trajectory of each initial cluster using Equation 2. We choose a hash table to store the information of the initial clusters for fast retrieval. More specifically, for each member of an initial cluster, the key-value pair of (tid, (refid, s)) is inserted into a hash table, where tid is an identifier of a trajectory, refid is an identifier of a representative trajectory, and s is the number of trajectories that belong to the initial cluster represented by refid. For example, for the initial cluster $c_1 = \{t_3, t_5, t_6\}$, we store key-value pairs such as (3, (5, 3)), (5, (5, 3)), and (6, (5, 3)) in the hash table, when t_5 is the representative trajectory of c_1 .

In addition, we also compute and store the distance relationship between all pairs of representative trajectories using Equation 1, which will reduce the number of distance computations at runtime. If there are *n* initial clusters, we build an $n \times n$ distance matrix DM, where DM_{ij} contains inter-cluster distances between c_i and c_j .

C. MergeDS

SemCluster invokes MergeDS when users issue Boolean + Clustering queries. This section presents the method by which MergeDS exploits the above pre-materialized summarized data

structure to merge initial clusters into meaningful groups. MergeDS consists of two components: 1) finding initial clusters satisfying given Boolean conditions and 2) merging them and maximizing the sum of inter-cluster distances.

Given a Boolean + Clustering query, we first search the database for trajectories satisfying the Boolean conditions. Then, we look up the hash table of the summarized data structure to identify the initial clusters of trajectories satisfying the Boolean conditions. Each identifier of a trajectory is used as a key. Using a hash table, we can effectively identify the initial clusters of trajectories that satisfy the Boolean conditions. Moreover, we can accelerate this step by linking leaf nodes of an index structure, which is built for a selection attribute, with the bins of the hash table.

For the initial clusters identified in the previous step, we find the disjoint clusters maximizing the inter-cluster distance D_{inter} , which is known to be NP-hard [22]. We develop a greedy approach, which continues merging a pair of clusters in ascending order of inter-cluster distance until a stop condition is satisfied. Note that the inter-cluster distance has been previously computed and stored in the distance matrix of the summarized data structure. Therefore, SemCluster does not require distance computations for all pairs of representatives but instead refers to the distance matrix for determining the order of merging clusters.

In each merging step, e.g., c_i and c_j , SemCluster elects a trajectory as a new representative trajectory of merged cluster $c_i \cup c_j$. Between two representatives of c_i and c_j , the representative from the larger cluster is promoted to be a new representative of $c_i \cup c_j$. To this end, we store the number of member trajectories in each merging step.

As a stop condition, we use the DB index [23], which is a quality measure validating the clustering results. Given a set of clusters $C = \{c_1, \dots, c_n\}$, the DB index is defined as:

$$DB(C) = \frac{1}{n} \sum_{i=1}^n R_i, \tag{10}$$

where $R_i = \max_{j=1, \dots, n, j \neq i} \frac{s_i + s_j}{d_{ij}}$, d_{ij} is the distance between cluster c_i and cluster c_j , and s_i is the variance of cluster c_i . The DB index approaches 0 as the clusters get more compact and well-separated. We note that the computation of the DB index does not require any additional distance computation between trajectories, which incurs high costs. We store the value of the DB index at each merging step. Then, we report the clustering result that minimizes the DB index.

VI. EXPERIMENTAL EVALUATION

We report the experimental results conducted to verify the effectiveness and efficiency of SemCluster for semantic-rich trajectories.

A. Experimental Settings

For semantic-rich trajectories, we collected geo-tagged images from the Flickr website using Flickr API. There were 1, 505, 768 geo-tagged images, which were taken between 1st January 2001 and 31st December 2009 in Australia. The collected

images were first grouped according to their user ID, and then the images of each user were sorted in order of time taken. We then split the image sequence of each user into disjoint subsequences by discretizing the 24-hour period into 12 range groups. As a result, 314, 000 semantic-rich trajectories are stored in the database.

All experiments were conducted on a machine with a Pentium-4 (3.2 GHz) CPU and 1 GBytes of main memory, running a version of the Linux operating system. We implemented our proposed algorithms using the C programming language. All trajectories were stored in an Oracle Berkeley DB (Version 4.6). The B-tree index was built on the database for efficient retrieval of Boolean selections.

B. Efficiency

In this section, we validate the efficiency of SemCluster in comparison with NNCluster [7] using the response time as a metric, by varying the selectivity of the Boolean conditions. Note that in contrast to NNCluster, SemCluster does not require

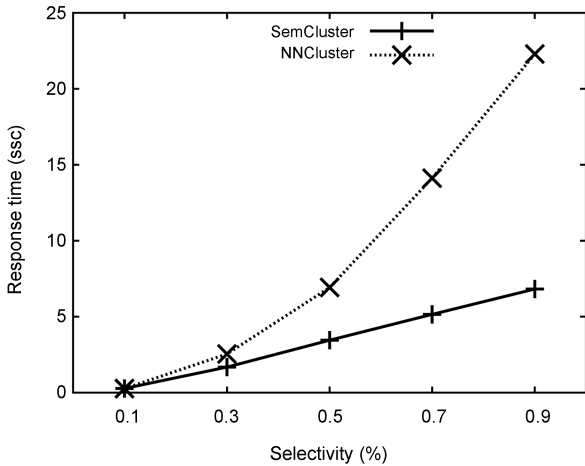


Fig. 3. Response time vs. selectivity.

any distance computations at runtime. Therefore, the performance gap between NNCluster and SemCluster will increase, if an expensive semantic-aware distance measure is used (e.g., computing the image similarity).

As shown in Fig. 3, SemCluster outperforms NNCluster in all experimental settings. The speedup of SemCluster increases as the selectivity grows. We achieve this speedup by pre-computing the trajectory distance computation when building the summarized data structure, and only handling references to inter-cluster distance in the distance matrix at runtime.

C. Quality of the Clustering Result

In this section, we validate the quality of the clustering results produced by SemCluster. More specifically, we visualize the clustering results for Boolean selections of timestamps or user ID.

A) Selection of Temporal Attributes: Scenarios for combining clustering with selection conditions of timestamps include finding clusters for different times of day or different seasons of the year. In this section, we demonstrate the clustering results for trajectories in the morning (S1), during lunchtime (S2) and in the evening (S3). Fig. 4a and 4b contrast results for S1 and S3—Note the cluster identified for S3 (evening) shows night view spots near the coastline, while the results for S1 (morning) includes more locations in downtown.

B) Selection of User ID: Selection conditions can be added to user ID to identify user-specific trends. We observe user-specific trends in the clustering result of semantic-rich trajectories. We use Flickr user ID as a selection attribute.

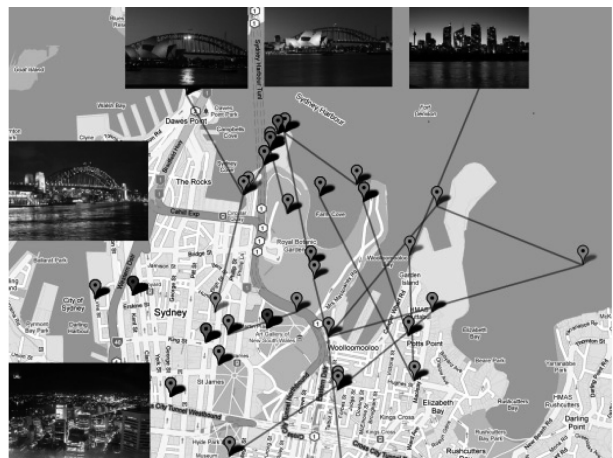
Fig. 5 illustrates trajectories satisfying the following Boolean condition:

- S4. : user ID = 2905XXXX@N02
- S5. : user ID = 2906XXXX@N04.

We also present the clustering result in Fig. 5 for each Boolean condition. For the purposes of presentation, we plot only the representative trajectory of clusters and we do not visualize small clusters with sizes less than 15% of the largest one. As



(a) Morning (6 am-12 pm)



(b) Evening (6 pm-12 am)

Fig. 4. Daily trend of semantic-rich trajectories.







BC	Trajectories satisfying BC	Clusters (representative trajectory only)	
S4	 68 trajectories	 C ₁ (30 trajectories)	 C ₂ (22 trajectories)
S5	 288 trajectories	 C ₁ (229 trajectories)	 C ₂ (31 trajectories)

Fig. 5. Clustering results of semantic-rich trajectory for Boolean conditions S4 and S5.

shown in Fig. 5, the clustering results for S4 and S5 show the user-specific trends in semantic-rich trajectories.

VII. CONCLUSION

In this paper, we studied a method to integrate a Boolean selection with clustering for mining semantic-rich trajectories. Specifically, we designed a new framework pre-materializing a summarized data structure for the whole dataset, and proposed a greedy clustering algorithm building on this structure. We validated the effectiveness and the efficiency of our results using real-life location trajectories and semantic-rich trajectories collected from the Flickr website. Our proposed framework enables us to capture temporal characteristics of clusters and outperforms the baseline algorithm proposed in our preliminary work. For future work, we are investigating the efficient maintenance of the summarized data structure, in case of frequent updates in the trajectory database.

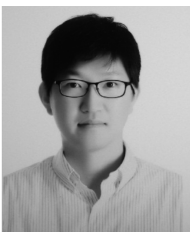
ACKNOWLEDGEMENTS

This work was supported by the Ministry of Knowledge Economy (MKE), Korea under the Information Technology Research Center (ITRC) support program supervised by the National IT Industry Promotion Agency (NIPA-2011-C1090-1131-0009).

REFERENCES

1. C. S. Jensen, D. Lin, and B. C. Ooi, "Query and update efficient B⁺-tree based indexing of moving objects," *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, Toronto, Canada, 2004, pp. 768-779.
2. H. Hu, J. Xu, and D. L. Lee, "A generic framework for monitoring continuous spatial queries over moving objects," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Baltimore, MD, 2005, pp. 479-490.
3. X. Xiong, M. F. Mokbel, and W. G. Aref, "SEA-CNN: scalable processing of continuous K-nearest neighbor queries in spatio-temporal databases," *Proceedings of the 21st International Conference on Data Engineering*, Tokyo, Japan, 2005, pp. 643-654.
4. S. Chen, B. C. Ooi, K. L. Tan, and M. A. Nascimento, "ST2B-tree: a self-tunable spatio-temporal B⁺-tree index for moving objects," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Vancouver, BC, 2008, pp. 29-42.
5. V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Collaborative location and activity recommendations with GPS history data," *Proceedings of the 19th International World Wide Web Conference*, Raleigh, NC, 2010, pp. 1029-1038.
6. Y. Zheng, L. Zhang, X. Xie, and W. Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," *Proceedings of the 18th International Conference on World Wide Web*, Madrid, Spain, 2009, pp. 791-800.
7. G. P. Roh and S. W. Hwang, "NNCluster: an efficient clustering algorithm for road network trajectories," *Database Systems for*

- Advanced Applications. Lecture Notes in Computer Science Vol. 5982*, H. Kitagawa, Y. Ishikawa, Q. Li, and C. Watanabe, Eds., Heidelberg, Germany: Springer Berlin, 2010, pp. 47-61.
8. J. Lou, Q. Liu, T. Tan, and W. Hu, "Semantic interpretation of object activities in a surveillance system," *Proceedings of the 16th International Conference on Pattern Recognition*, Quebec, QC, Canada, 2002, pp. 777-780.
 9. I. N. Junejo, O. Javed, and M. Shah, "Multi feature path modeling for video surveillance," *Proceedings of the 17th International Conference on Pattern Recognition*, Cambridge, UK, 2004, pp. 716-719.
 10. J. Gudmundsson, M. v. Kreveld, and B. Speckmann, "Efficient detection of motion patterns in spatio-temporal data sets," *Proceedings of the 12th annual ACM international workshop on Geographic information systems*, Washington DC, 2004, pp. 250-257.
 11. H. Cao and O. Wolfson, "Nonmaterialized motion information in transport networks," *Database Theory: ICDT 2005. Lecture Notes in Computer Science Vol. 3363*, T. Eiter and L. Libkin, Eds., Heidelberg, Germany: Springer Berlin, 2005, pp. 173-188.
 12. J. G. Lee, J. Han, and K. Y. Whang, "Trajectory clustering: a partition-and-group framework," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Beijing, China, 2007, pp. 593-604.
 13. B. Lin and J. Su, "One way distance: for shape based similarity search of moving object trajectories," *GeoInformatica*, vol. 12, no. 2, pp. 117-142, 2008.
 14. M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," *Proceedings of the 18th International Conference on Data Engineering*, San Jose, CA, 2002, pp. 673-684.
 15. L. Chen and R. Ng, "On the marriage of Lp-norms and edit distance," *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, Toronto, Canada, 2004, pp. 792-803.
 16. L. Chen, M. T. Ozsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Baltimore, MD, 2005, pp. 491-502.
 17. H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen, "Discovery of convoys in trajectory databases," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 1068-1080, 2008.
 18. S. Gaffney and P. Smyth, "Trajectory clustering with mixtures of regression models," *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, 1999, pp. 63-72.
 19. C. Li, M. Wang, L. Lim, H. Wang, and K. C. C. Chang, "Supporting ranking and clustering as generalized order-by and group-by," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Beijing, China, 2007, pp. 127-138.
 20. S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, "On map-matching vehicle tracking data," *Proceedings of the 31st International Conference on Very Large Data Bases*, Trondheim, Norway, 2005, pp. 853-864.
 21. R. M. Karp, "Reducibility among combinatorial problems," *Complexity of Computer Computations: Proceedings*, R. E. Miller, J. W. Thatcher, United States Office of Naval Research Mathematics Program, IBM World Trade Corporation, and International Business Machines Corporation Mathematical Sciences Department, Eds., New York, NY: Plenum Press, 1972, pp. 85-103.
 22. Y. Leung, J. S. Zhang, and Z. B. Xu, "Clustering by scale-space filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1396-1410, 2000.
 23. D. L. Davies and D. W. Bouldin, "Cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224-227, 1979.



Gook-Pil Roh

Gook-Pil Roh is a graduate student of Computer Science and Engineering at POSTECH.



Seung-won Hwang

Seung-won Hwang is an associate professor of Computer Science and Engineering at POSTECH.