Exploration of Interesting Dense Regions on Spatial Data

Double-blind submission

ABSTRACT

The large size of spatial data hinders its effective analysis for insight discovery. Analysts require to obtain few high quality options (i.e., "highlights") to have a more focused investigation. In this paper, we define, formalize and analyze Interesting Dense Regions (IDRs) which capture implicit preferences of analysts in order to automatically find highlights. Our approach involves a polygon-based abstraction layer for captured preferences. Using these interesting dense regions, we highlight points to guide the analysis process. We analyze the efficiency and effectiveness of our approach through a realistic example.

KEYWORDS

Spatial data, Implicit feedback, Dense regions, Spatial highlights.

ACM Reference Format:
Double-blind submission. 2018. Exploration of Interesting Dense Regions on Spatial Data. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM'18),* Rakesh Agrawal, Andrei Broder, and Mohammed Zaki (Eds.). ACM, New York, NY, USA, Article 4, 4 pages. https://doi.org/10.475/XXX_X

1 INTRODUCTION

Spatial data are often voluminous. Hence the focus in the literature of spatial data analysis is often on "efficiency", i.e., enabling fluid means of navigation in spatial data in order to facilitate the exploratory analysis. The common approach is to design pre-computed indexes which enable efficient retrieval of spatial data (e.g., [13]). However, there has been fewer attention to the "value" derived from spatial data. Despite the huge progress on the efficiency front, an analyst may easily get lost in the plethora of geographical points due to two following reasons.

- In an exploratory context, the analyst doesn't know apriori what to investigate next.
- Moreover, she may easily get distracted and miss interesting points by visual clutter caused by huge point overlaps.

The main drawback of the traditional analysis model (e.g., [12, 14], to name a few) is that the analyst has a *passive role* in the process. In other words, the analyst's feedback (i.e., her likes and dislikes) is ignored and only the input query (i.e., her explicit request) is served. In case feedback is incorporated, the process can be more directed towards analyst's interests where her partial needs can be served earlier in the process. In this paper, we advocate for a "guidance layer" on top of the raw visualization of spatial data to enable analysts know *"what to see next"*. This guidance should be a function of analyst feedback: the system should recommend options similar to what the analyst has already appreciated.

Considering this context, we propose in this paper an approach to explore Interesting Dense Regions (IDRs) whose aim is to capture and analyze implicit feedback of analysts in spatial data analysis. Without loss of generality, we focus on "mouse moves" as the implicit feedback received from the analyst. Mouse moves are the most common way that analysts interact with geographical maps [7]. However, our approach can be easily extended to other types of inputs such gaze tracking [2] and leap motions.

2 INTERESTING DENSE REGIONS

An Interesting Dense Region (IDR) is a spatial region with high probability of having spatial points with potential interest to the analyst. An IDR is captured and defined by processing user feedbacks. Unlike the literature which mainly focuses on explicit refections, we investigate on implicit feedback. There are different ways to capture implicit feedbacks.

- During spatial data analysis, it is often the case that analysts look at some regions of interest but forget to provide an explicit feedback. We call this latent signal, gaze. It shown in [2] that gaze has a strong correlation with "user attention". The signal can be captured by tracking eye movements.
- To address privacy issues of gaze exploitation, we consider an alternative option, i.e., tracking the mouse cursor. It is shown in [3] that mouse gestures have a strong correlation with "user engagement". Intuitively, a point receives a positive feedback if the cursor moves around it frequently.

The objective of discovering IDRs is to obtain the preferences of analysts which have never been expressed explicitly. For such discovery, we are inspired by two observations.

■ **Observation 1.** We believe that a region appeals more interesting to the analyst if it is denser, i.e., the analyst moves her mouse in that region several times.

■ **Observation 2.** It is possible that the analyst moves her mouse everywhere in the map. This should not signify that everywhere in the map has the same significance.

We consider two different layers on a geographical map: "spatial layer" and "interaction layer". The spatial layer contains points from a spatial database $\mathcal{P}$. The interaction layer contains mouse move points $\mathcal{M}$. Our proposed approach mines a set of mouse move points $\mathcal{M}$ in the interaction layer to discover one or several IDRs, in which most analyst's interactions occur. Then it matches the spatial points $\mathcal{P}$ with IDRs in order to find points inside each region. The attributes of resulting points will be exploited to update analyst's feedback vector F. The updated vector F will then be used to find k highlights (where k is specified by the analyst). These steps ensure that the final highlights reflect analyst's implicit interests.

Algorithm 1 summarizes our approach for mining IDRs. We add points to $\mathcal{M}$ only every $200ms$ to prevent adding redundant

CIKM'18, October 2018, Lingotto, Turin Italy

ACM ISBN XXX-XXX-XX-XXX/10/18.
https://doi.org/10.475/XXX_X

Algorithm 1: Find Interesting Dense Regions (IDRs)

Input: Current time t_c, mouse move points $\mathcal{M}$
Output: IDRs $\mathcal{S}$

1 $\mathcal{S} \leftarrow \emptyset$
2 $g \leftarrow$ *number of time segments*
3 **for** $i \in [0, g]$ **do**
4 $\quad \mathcal{M}_i \leftarrow \{m = \langle x, y, t \rangle | (\frac{t_c}{g} \times i) \leq t \leq (\frac{t_c}{g} \times (i+1))\}$
5 $\quad C_i \leftarrow$ *mine_clusters*$(\mathcal{M}_i)$
6 $\quad O_i \leftarrow$ *find_ploygons*(C_i)
7 **end**
8 **for** O_i, O_j *where* $i, j \in [0, g]$ *and* $i \neq j$ **do**
$\quad \mathcal{S}.append(intersect(O_i, O_j))$
9 **return** $\mathcal{S}$

points. Following Observation 1 and in order to mine the recurring behavior of the analyst, the algorithm begins by partitioning the set $\mathcal{M}$ into g fixed-length consecutive segments $\mathcal{M}_0$ to $\mathcal{M}_g$. The first segment starts at time zero (where the system started), and the last segment ends at t_c, i.e., the current time. Following Observation 2, we then find dense clusters in each segment of $\mathcal{M}$ using a variant of DB-SCAN approach [9]. Finally, we return intersections among those clusters as IDRs.

For clustering points in each time segment (i.e., line 5 of Algorithm 1), we use ST-DBSCAN, a space-aware variant of DB-SCAN for clustering points based on density [6]. For each subset of mouse move points $\mathcal{M}_i, i \in [0, g]$, ST-DBSCAN begins with a random point $m_0 \in \mathcal{M}_i$ and collects all density-reachable points from m_0 using a distance metric. As mouse move points occur in the 2-dimensional pixel space (i.e., the display), we choose euclidean distance as our metric. If m_0 turns out to be a core object, a cluster will be generated. Otherwise, if m_0 is a border object, no point is density-reachable from m_0 and the algorithm picks another random point in $\mathcal{M}_i$. The process is repeated until all of the points have been processed.

Once clusters are obtained for all subsets of $\mathcal{M}$, we find their intersections to locate recurring regions (line 6). To obtain intersections, we need to clearly define the spatial boundaries of each cluster. Hence for each cluster, we discover its corresponding polygon that covers the points inside. For this aim, we employ Quickhull algorithm, a quicksort-style method which computes the convex hull for a given set of points in a 2D plane [4].

There exist several approaches to infer a spatial region for a given set of points [1, 5, 8, 10, 11]. The common approach is to cluster points in form of concave and convex polygons. In case a concave polygon is constructed, the "dents" of such a polygon may entail points which are not necessarily in $\mathcal{M}$. In IDR algorithm, however, we adapt Quickhull [4], due its simplicity, efficiency and it's natural implementation of convex polygons.

3 CASE STUDY AND EXPERIMENTS

This section describes an example to guide the experiments made over the proposal presented in this paper. The following use case describes a scenario of implicit feedbacks.

Example. *Lucas is planning to pass some days in Paris, France. He loves French culture and language, and pass some days in a nice city would be delighted to him. He decides to rent a home-stay from Airbnb website*[1]*. He likes to discover the city, hence he is open to any type of lodging in any region with an interest to stay in the city center. The website returns 4000 different locations. As he has no other preferences, an exhaustive investigation needs scanning each location independently which is nearly infeasible. While he is scanning few first options, he shows interest in the region of Champ de Mars (close to Eiffel Tower), but he forgets or doesn't feel necessary to click a point there. An ideal system should capture this implicit feedback in order to short-list a small subset of locations that Lucas should consider as high priority.*

We describe the process of finding IDRs in an example. Figure 1 shows the steps that Lucas follows in our running example to explore home-stays in Paris. Figure 1.A shows mouse movements of Lucas in different time stages. In this example, we consider g = 3 and capture Lucas feedback in three different time segments (progressing from Figures 1.B to 1.D). It shows that Lucas started his search around Eiffel Tower and Arc de Triomphe (Figure 1.B) and gradually showed interest in south (Figure 1.C) and north (Figure 1.D) as well. All intersections between those clusters are discovered (hatching regions in Figure 1.E) which will constitute the set of IDRs (Figure 1.F), i.e., IDR1 to IDR4.

In the Airbnb dataset points are home-stays which are shown with their nightly price on the map. We observe (Figure 2) that there exist many matching points with IDR3 and absolutely no matching point for IDR2. For IDR4, although there exist many home-stays below the region, we never check their containment, as they belong to a Quadtree cell which doesn't intersect with the IDR.

Considering the described example we perform two sets of experiments. A first set of few experiments to validate its efficiency and effectiveness. The second set of experiments aims to analyse the number of IDRs generated for each interaction, the number of points and its percentage highlighted from the dataset. In the interest of space, we only present a glimpse of our experiments here. More will be discussed in an extended version.

First off, we validate the "usability" of our proposal. For this aim, we design a user study with some participants who are all students of Computer Science. Some of them are "novice" users who don't know the location, and some are "experts". Participants should fulfill a task. For each participant, we report a variant of time-to-insight measure, i.e., how long the participants interact with the frameworks before fulfilling the task. Evidently, less number of interactions is preferred as it means that the participant can reach insights faster. Then we measure the time to execute the task.

On the Airbnb dataset of Paris with 1,000 points, we define two different tasks: *T1: "finding a point in a requested location"* (e.g., find a home-stay in the "*Champ de Mars*" area), and *T2: "finding a point with a requested profile"* (e.g., find a cheap home-stay.) Participants may also begin their navigation either from *I1: "close to the goal"* or *I2: "far from the goal"*.

Participants benefit from information highlighting based on their implicit feedback. Table 1 reports the time for the interactions to find what they are looking for, novice and expert participants,

[1] *http://www.airbnb.com*

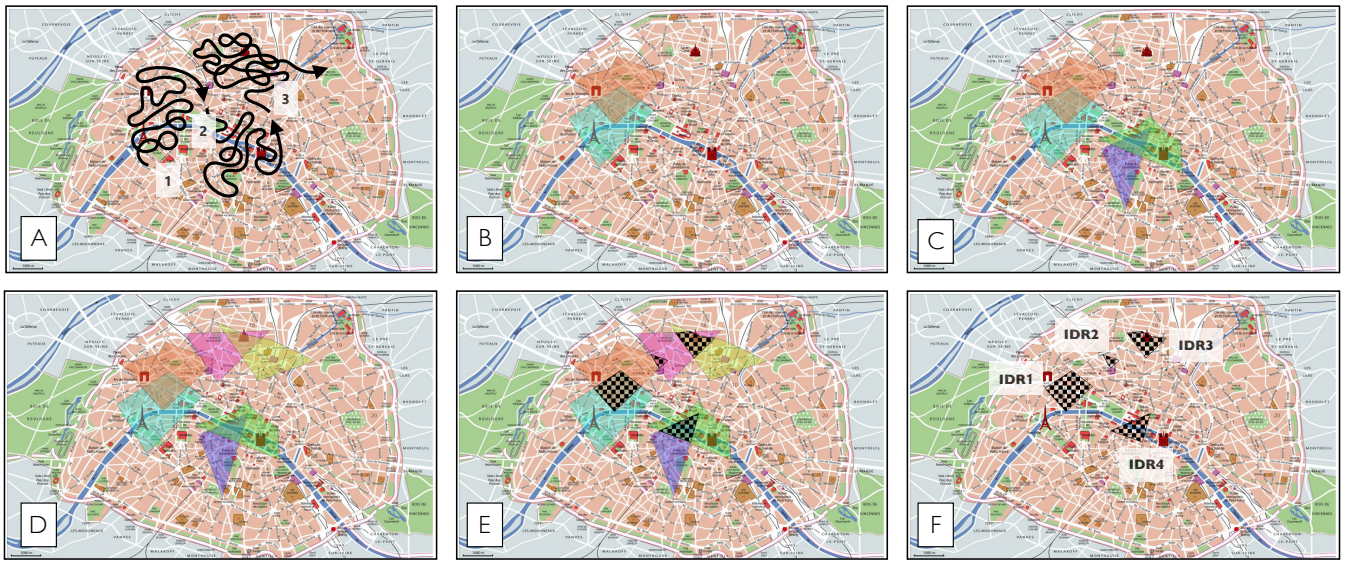

Figure 1: The process of finding IDRs - Paris city.

Figure 2: Matching points for IDRs.

Table 1: "Novice" and "Experts" - Interactions (in seconds)

	T1/I1	T2/I1	T1/I2	T2/I2
Novices	1.99	2.38	2.00	2.48
Experts	1.72	2.09	1.70	2.14

respectively. We observe that on average the system spends 2.067 seconds to return a defined goal. This shows that implicit feedback capturing is an effective mechanism which helps analysts to reach their goals faster.

Expert participants need 0.35 (seconds) fewer interactions on average. Interestingly, starting points, i.e., *I1* and *I2*, do not have a huge impact on number of steps. It is potentially due to the diversity component which provides distinct options and can quickly guide analyst towards their region of interest. We also observe that the task *T2* is an easier task than *T1*. This is potentially due to where the analyst can request options similar to what she has already observed and greedily move to her preferred regions.

The second set of experiments consider 2 different datasets, Airbnb [2] and Yelp [3] datasets with spatial points in Paris city, apartment and restaurant datasets, respectively. For each type of dataset we proceed the experiments with 4 volume of points, 100, 1000, 2000 and 4000 points, respectively. For each volume of points we proceed 20 interactions, and we present the results averages. So, we proceed 80 interactions using the Airbnb dataset and 80 interactions using the Yelp dataset.

We pass 2 minutes trying to find an interesting point, in each interact. We capture, in each interaction, the following informations: (i) the number of regions created from the mouse movements during the interactions; (ii) the number of IDRs (intersection of regions) generated after the interactions; (iii) the number of points from the dataset presented in each IDR after the interaction; and finally (iv) the percentage of points (from the dataset volume) inside the IDRs.

In this set of experiment we validate not only the "usability" of our proposal, but also we analyse how the IDRs are generated and the how many interesting regions and points can be highlighted from the implicit feedbacks captured during the user interactions. Tables 2 and 3 present the result for each dataset.

[2] https://www.airbnb.com/ and http://insideairbnb.com/get-the-data.html
[3] https://www.yelp.com/

For the apartment dataset experiments (Table 2) the number of regions decreases while the number of points increases. The general average of regions created is around 10, per interaction. The same happens for IDRs, except when we have 4000 points. The average of points presented in IDRs is 24.97. So, the system is able to highlight at least 8.05% of point from the dataset, in average.

Table 2: Airbnb dataset - Paris city

n. points	regions	IDRs	points in IDRs	% points
100	11.35	10.05	29.40	29.40%
1000	10.75	6.75	11.70	1.17%
2000	7.37	3.63	5.63	0.003%
4000	10.30	10.15	53.15	1.33%
average	**9.94**	**7.64**	**25.97**	**8.05%**

The results for the restaurant dataset (Table 3) is more uniform than the one presented in Table 2. The general average of regions created is 12.75 regions per interaction. The number of regions also decreases while the number of points increases. The same happens for IDRs, where we have an average of 8.9 IDRs generated per interaction. The number of points presented in IDRs is 108.65, in average, and it represents 13.11% of points highlighted from the dataset, in average.

For both datasets we considered the location presented in our example scenario, where Lucas is planning to pass some vacations days in Paris.

Table 3: Yelp dataset - Paris city

n. points	regions	IDRs	points in IDRs	% points
100	14.90	7.55	28.30	28.30%
1000	13.90	10.00	149.55	14.96%
2000	11.05	9.80	111.05	5.55%
4000	10.45	8.55	145.7	3.64%
average	**12.57**	**8.97**	**108.65**	**13.11%**

4 CONCLUSIONS

In this paper, we propose to explore Interesting Dense Regions (*IDRs*) of spatial information highlighting using implicit feedback. The implicit feedbacks are captured from mouse movements of the analyst over the geographical map. We formulate and formalize a novel polygon-based capturing algorithm which returns few highlights (regions) in line with analyst's implicit preferences.

We consider some directions of improvement for this work. We are interested to incorporate an "explainability" component which can describe causalities behind preferences. For instance, we are interested to find seasonal patterns to see why the preferences of analysts change from place to place during various seasons of the year. Another direction is to incorporate "Query by Visualization" approaches, where analysts can specify their intents alongside their implicit preferences, directly on the map [15].

REFERENCES

[1] Avi Arampatzis, Marc van Kreveld, Iris Reinbacher, Christopher B. Jones, Subodh Vaid, Paul Clough, Hideo Joho, and Mark Sanderson. 2006. Web-based delineation of imprecise regions. *Computers, Environment and Urban Systems* 30, 4 (2006), 436 – 459. https://doi.org/10.1016/j.compenvurbsys.2005.08.001 Geographic Information Retrieval (GIR).

[2] Ioannis Arapakis, Mounia Lalmas, B Barla Cambazoglu, Mari-Carmen Marcos, and Joemon M Jose. 2014. User engagement in online news: Under the scope of sentiment, interest, affect, and gaze. *Journal of the Association for Information Science and Technology* 65, 10 (2014), 1988–2005.

[3] Ioannis Arapakis, Mounia Lalmas, and George Valkanas. 2014. Understanding within-content engagement through pattern analysis of mouse gestures. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 1439–1448.

[4] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. 1996. The Quickhull Algorithm for Convex Hulls. *ACM Trans. Math. Softw.* 22, 4 (Dec. 1996), 469–483. https://doi.org/10.1145/235815.235821

[5] Michael Bevis and Jean-Luc Chatelain. 1989. Locating a point on a spherical surface relative to a spherical polygon of arbitrary shape. *Mathematical Geology* 21, 8 (01 Oct 1989), 811–828. https://doi.org/10.1007/BF00894449

[6] Derya Birant and Alp Kut. 2007. ST-DBSCAN: An Algorithm for Clustering Spatial-temporal Data. *Data Knowl. Eng.* 60, 1 (Jan. 2007), 208–221. https://doi.org/10.1016/j.datak.2006.01.013

[7] Mon Chu Chen, John R. Anderson, and Myeong Ho Sohn. 2001. What Can a Mouse Cursor Tell Us More?: Correlation of Eye/Mouse Movements on Web Browsing. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems (CHI EA '01)*. ACM, New York, NY, USA, 281–282. https://doi.org/10.1145/634067.634234

[8] Matt Duckham, Lars Kulik, Mike Worboys, and Antony Galton. 2008. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition* 41, 10 (2008), 3224 – 3236. https://doi.org/10.1016/j.patcog.2008.03.023

[9] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*. AAAI Press, 226–231. http://dl.acm.org/citation.cfm?id=3001460.3001507

[10] M.J. Fadili, M. Melkemi, and A. ElMoataz. 2004. Non-convex onion-peeling using a shape hull algorithm. *Pattern Recognition Letters* 25, 14 (2004), 1577 – 1585. https://doi.org/10.1016/j.patrec.2004.05.015

[11] Antony Galton and Matt Duckham. 2006. What Is the Region Occupied by a Set of Points?. In *Geographic Information Science*, Martin Raubal, Harvey J. Miller, Andrew U. Frank, and Michael F. Goodchild (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 81–98.

[12] Niranjan Kamat, Prasanth Jayachandran, Karthik Tunga, and Arnab Nandi. 2014. Distributed and interactive cube exploration. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*. IEEE, 472–483.

[13] Lauro Lins, James T Klosowski, and Carlos Scheidegger. 2013. Nanocubes for real-time exploration of spatiotemporal datasets. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2456–2465.

[14] Behrooz Omidvar-Tehrani, Sihem Amer-Yahia, and Alexandre Termier. 2015. Interactive User Group Analysis. In *CIKM*. ACM, 403–412. https://doi.org/10.1145/2806416.2806519

[15] Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios, and Aditya Parameswaran. 2016. Effortless data exploration with zenvisage: an expressive and interactive visual analytics system. *Proceedings of the VLDB Endowment* 10, 4 (2016), 457–468.