GeoGuide: An Interactive Guidance Approach for Spatial Data

Behrooz Omidvar-Tehrani[†], Plácido A. Souza Neto[‡]
[†]The Ohio State University, USA, [‡]Federal Institute of Rio Grande do Norte - IFRN, Brazil
[†]omidvar-tehrani.1@osu.edu, [‡]placido.neto@ifrn.edu.br

ABSTRACT

spatial data is becoming increasingly available in various domains such as transportation and social science. Discovering patterns and trends in this data provides improved insights for planning and decision making for smart city management, disaster management and other applications. However, exploratory analysis of such data is a challenge due to its huge size and diversity of spatial data. It is often unclear for the analyst *what to see next* during an analysis process, i.e., lack of guidance. To tackle this challenge, we formulate guidance as an optimization problem and develop GeoGuide, an efficient interactive guidance approach for spatial data. At each step of an interactive process, k-most interesting geographical points become highlighted to guide the analyst through further steps. We illustrate the efficiency and usability of our framework in an extensive set of experiments.

1. INTRODUCTION

Nowadays, there exists huge amounts of spatial data in various fields of science, such as agriculture, transportation and social science. Analysis of such data is interesting as it is grounded on reality: each record represents a specific geographical location. Moreover, understanding patterns and trends provides insights leading to improved user planning and decision making. Some instance applications of spatial data are smart city management, disaster management and autonomous transport.

Spatial data analysis is often performed in *exploratory context*: the analyst does not have a precise query in mind and she explores data in iterative steps in order to find potentially interesting results. Traditionally, an exploratory analysis scenario on spatial data is described as follows: the analyst visualizes a subset of data using a query in an off-the-shelf product (e.g., Tableau[1], Spotfire[2]). The result will be illustrated on a geographical map. Then she investigates on different parts of the visualization by zooming in/out and panning the map in order to discover patterns and trends of interest. The analyst may iterate on this process several times by issuing different queries and focusing on different aspects of data.

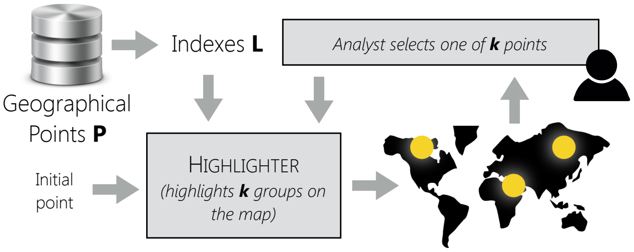
Geographical Points **P**
Indexes **L**
Analyst selects one of ***k*** points
Initial point
Highlighter
(highlights ***k*** groups on the map)

Figure 1: GeoGuide Framework

The literature in spatial data analysis has a focus on *efficiency* of exploratory iterations: *"how can analysts navigate in spatial data fluidly?"* The common approach is to design pre-computed indexes which enable efficient retrieval of spatial data (e.g., [5]). However, there has been fewer attention to the *value* of spatial data. Despite the huge progress on efficiency front, an analyst may easily get lost in the plethora of geographical points because *i.* she doesn't know what to investigate next in an exploratory context and *ii.* she may get distracted and miss interesting points by visual clutter caused by huge point overlaps. In other words, although iteration transitions (between one analysis step to the other) can be performed efficiently, the decision which forms a transition, remains unclear. The following example illustrates the challenge in practice.

Example 1. *Liam is planning a short trip to Paris. He decides to rent a home-stay from Airbnb website[3]. He is open to any type of lodging and he wants to explore different options (i.e., exploratory analysis). He queries all available locations in Paris with a fair price. His query results in 3000 locations. As he has no other preferences, an exhaustive investigation needs scanning each location independently which is nearly infeasible. In case he wants to focus on a smaller set of options, it is not clear which subset he needs to look at. While he is looking at primary locations in the list, he shows interest in having "balcony" as amenity and being close to Eiffel tower. An ideal system can capture this feedback in*

[1] *http://www.tableau.com*
[2] *http://spotfire.tibco.com*
[3] *http://www.airbnb.com*

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-nd/4.0/. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org.
Proceedings of the VLDB Endowment, Vol. 10, No. 6
Copyright 2017 VLDB Endowment 2150-8097/17/02.

order to short-list a small subset of remaining locations that Liam should consider as high priority.

To overcome the challenge of value in exploratory analysis, visualization environments offer a complete tool-set to manipulate data (filter, aggregate, etc.). In practice, this duplicates the problem: the analyst is left alone in a huge space of spatial data and tools. The principled challenge for the analyst is *"what to see next"* in the exploratory context. A *guidance* mechanism is then necessary to point out potential future directions of analysis.

In this demo paper, we introduce GeoGuide, an interactive framework to highlight a subset of geographical points based on analyst feedback. The highlighted set facilitates the decision-making process by providing guidance on what the analyst should potentially concentrate on. The set of highlights is deliberated over high quality. We consider two quality metrics in GeoGuide: *relevance* and *diversity*. First, each highlighted point should be relevant to historical choices of the analyst. Second, highlights should be geographically diverse to unveil different aspects of analysis. Both quality metrics are interdependent to compute the set of highlights.

The literature contains several instances of feedback exploitation to guide the analyst in further analysis steps (e.g., [2]). The common approach is a top-k processing methodology in order to prune the search space based on the explicit feedback and recommend a small subset of interesting results of size k. A clear distinction of GeoGuide is that it doesn't aim for pruning, but leveraging the actual data with potential interesting results that the analyst may miss due to the huge volume of spatial data. While in top-k processing algorithms, analyst choices are limited to k, GeoGuide has a freedom of choice where highlights get seamlessly updated with new analyst choices. The following example describes an application of GeoGuide in business domain.

Example 2. *Tiffany is a data scientist and is tasked to design a "chain marketing" strategy for a Peking Duck product (a Chinese duck dish). She decides to exploit Yelp data*[4] *(i.e., restaurant check-ins) to find out the advertisement chain. She performs her analysis in* GeoGuide. *In the first step, she shows interest towards New York region, where the headquarters of the company is located and the product has already gained success. The system will then provide few highlights in diverse regions: San Fransisco, Washington DC and Marlton, NJ. All regions seem interesting to Tiffany as they exhibit similar eating profile with New York, hence potentials for chaining the advertisement. She decides to pick Marlton due to its proximity so that she can reduce transportation costs. The system will then provide other highlights based on her updated feedback. She can then make the city-to-city chain marketing strategy in iterative steps using highlights.*

There exist few instances of information-highlighting methods in the literature [4, 7, 9, 8]. However all these methods are *objective* and do not apply to the context of spatial guidance where user feedback is involved. In terms of recommendation, few approaches focus on spatial dimension [1, 3] while the context and result diversification are missing.

2. GEOHIGHLIGHT FRAMEWORK

In this section, we describe the functionality of GeoGuide which is an inspiration from both recommendation [6] and visual highlighting [4, 7] methodologies. The aim of this framework is to guide analysts in large-scale spatial data analysis. We follow a highlighting strategy where we stress out regions of interest based on analyst historical choices (i.e., feedback).

In our framework, we consider a spatial database $\mathcal{D}$ consisting $\langle \mathcal{P}, \mathcal{A} \rangle$ where $\mathcal{P}$ is the set of geographical points and $\mathcal{A}$ is the set of point attributes. For each $p \in \mathcal{P}$, we consider a tuple $\langle lat, lon, alt \rangle$ which denotes p's geographical coordinates (latitude, longitude and altitude respectively). The set $\mathcal{A}_p$ contains attribute-values for p over the schema of $\mathcal{A}$. For instance, on a bike-sharing dataset, $\mathcal{A}_p = \langle \texttt{female}, \texttt{young}, \texttt{hybrid-bike} \rangle$ on the schema $\mathcal{A} = \langle \texttt{gender}, \texttt{age}, \texttt{type} \rangle$ denotes that p is associated to a young female cyclist who rides a hybrid bike. The set $\mathcal{A}$ is domain-dependent and defines the semantics of a spatial dataset.

We also define a feedback vector $\mathcal{F}$ on the schema $\mathcal{A}$ initialized by zero. The vector gets updated by $\mathcal{A}_p$ whenever the analyst shows interest in a geographical point p. Feedback vector is always kept normalized where $\Sigma_{v \in \mathcal{F}}(v) = 1.0$. Unlike the literature which mainly focuses on explicit feedback (where the analyst should clearly reflect her likes and dislikes), we investigate on implicit feedback. This enables the system to capture *what the analyst may miss* instead of what the analyst has clearly investigated before. We consider two different ways to capture implicit feedback.

- **Gaze.** During spatial data analysis, it is often the case analysts look at some regions of interest but forget to provide an explicit feedback. For instance in Example 1, while Liam is focusing on home-stays close to the Eiffel tower, he also looks at farther locations with easy train access. However, he never clicks on them. We call this latent signal, *gaze*. Gaze can be captured by tracking the directions of analyst looking. It can be done for example by http://www.xlabsgaze.com/. Also we can imitate by cursor tracking.

- **Session Time.** In most spatial datasets, there is profile page dedicated to each point. For instance restaurant pages in Yelp. We measure the amount of time that the user takes in a page as implicit feedback. For instance, if the user looks a lot at Indian page, then it seems that he is interested in this type of resturants.

At each step of the analysis, GeoGuide highlights few points based on the $\mathcal{F}$ content. The decision is made based on two quality metrics, i.e., relevance and diversity.

Relevance. Highlights should be in the same line with previous choices of the analyst. Note that we consider *contextual-based* relevance and not *distance-based* relevance. The reason originates from our data observation. For instance in a taxi dataset, consider a ride in New York for a young male customer for an itinerary of 10 kilometers and \$3 tip. In contrary to thousands of kilometers of geographical distance, the ride is very relevant to another one in San Fransisco for a middle-age male customer for an itinerary of 8 kilometers and \$2.5 tip.

The relevance between a point p and the feedback vector $\mathcal{F}$ is defined as follows.

[4] *https://www.yelp.com/*

$$relevance(p, \mathcal{F}) = average_{a \in \mathcal{A}_p \cap \mathcal{F}}(sim(p, \mathcal{F}, a)) \quad (1)$$

The similarity function *sim*() can be any function such as Jaccard and Cosine. Each attribute can have its own similarity function (as string and integer attributes are compared differently.) Then *sim*() works as an overriding-function which provides encapsulated similarity computations for any type of attribute.

The point-feedback relevance is neither monotonic nor anti-monotonic. In other words, a highly relevant point to the feedback vector at step i may become totally irrelevant to the updated feedback vector at step $i + 1$. The reason is the dynamic nature of the feedback vector which may drastically evolve at each analysis step. For efficiency reasons, we employ a static component and build indexes which capture point-point relevance to speed up online computations.

Diversity. Highlighted points should also represent distinct regions so that the analyst can observe different aspects of data and decide based on the big picture. Given a set of points $s = \{p_1, p_2 \ldots\}$, we define *diversity* as follows.

$$diversity(s) = average_{\{p,p'\} \subseteq s | p \neq p'} distance(p, p') \quad (2)$$

The function $distance(p, p')$ operates on geographical coordinates of p and p' and can be considered as any distance function of Chebyshev distance family such as Eucledian. However, as distance computations are done in *spherical space* using latitude, longitude and altitude, it is au-naturel to employ Harvestine distance shown in Equation 3.

$$\begin{aligned} distance(p, p') = [&acos(cos(p_{lat}).cos(p'_{lat}).cos(p_{lon}).cos(p'_{lon}) \\ &+ cos(p_{lat}).sin(p'_{lat}).cos(p_{lon}).sin(p'_{lon}) \\ &+ sin(p_{lat}).sin(p'_{lat}))] \times earth\_radius \end{aligned} \quad (3)$$

GeoGuide requires an efficient algorithm for dynamically analyzing and comparing geographical points (Figure 1). Although GeoGuide operates on points, its functionality can be easily extended to regions using point-clustering methods such as k-means. Intuitively, GeoGuide is a layer on top of raw visualization which highlights representatives at each analysis iteration. The representatives reflect the feedback of the analyst in previous steps.

GeoGuide operates in two steps: Preparation and Highlighter. In order to speed up computing relevance in online execution, we pre-compute an inverted index for each single geographical point in $\mathcal{P}$ in the offline Preparation step (as is commonly done in Web search). Each index $\mathcal{L}_p$ for the point p stores all other points in $\mathcal{P}$ in decreasing order of their relevance with p. Thanks to the parameter σ, we only partially materialize the indexes.

Algorithm 1 illustrates the online execution step of GeoGuide so called Highlighter. The algorithm is a single greedy procedure that solves the GeoGuide problem. Highlighter is called at each interactive step of GeoGuide (as in Figure 1). The algorithm admits as input a point $p \in \mathcal{P}$ and returns the best k points denoted $\mathcal{S}_p$.

To comply with the desiderata **D5**, we consider a time limit parameter *tlimit* in Algorithm 1. In a *best-effort* strategy, the algorithm bounds user waiting time by *tlimit* to return the best possible results by then.

Highlighter begins by retrieving the most relevant points to p by simply retrieving the k highest ranking points in $\mathcal{L}_p$ (line 1). Function $get\_next(\mathcal{L}_p)$ (Line 2) returns the next point p_{next} in $\mathcal{L}_p$ in sequential order. Lines 3 to 11 iterate over the inverted indexes to determine if other points should be considered to increase diversity while staying within the time limit and not violating the relevance threshold with the selected point. Since points in $\mathcal{L}_g$ are sorted on decreasing relevance with p, the algorithm can safely stop as soon as the relevance condition is violated (or if the time limit is exceeded).

Algorithm 1: Highlighter Algorithm

```
   Input: $p \in \mathcal{P}$, $\sigma$, $k$, $tlimit$
1  $\mathcal{S}_p \leftarrow get\_top\_k(\mathcal{L}^p)$
2  $p_{next} \leftarrow get\_next(\mathcal{L}^p)$
3  while ($tlimit$ not exceeded $\wedge$ $relevance(p, p_{next}) \geq \sigma$) do
4      for $p_{current} \in \mathcal{S}_p$ do
5          if $diversity\_improved(\mathcal{S}_p, p_{next}, p_{current})$ then
6              $\mathcal{S}_p \leftarrow replace(\mathcal{S}_p, p_{next}, p_{current})$
7              break
8          end
9      end
10     $p_{next} \leftarrow get\_next(\mathcal{L}^p)$
11 end
12 return $\mathcal{S}_p$
```

The algorithm then looks for a candidate point $p_{current} \in \mathcal{S}_p$ to replace in order to increase diversity. The boolean function *diversity_improved*() (line 5) checks if by replacing $p_{current}$ by p_{next} in $\mathcal{S}_p$, the overall diversity of the new $\mathcal{S}_p$ increases.

At each iteration, the feedback vector is compared with all points to find the ones which are more similar to analyst feedback. To speed up comparison, we exploit boolean representations. We convert both the feedback and point identity to booleans and make boolean comparisons which are indeed faster.

We dont want to be slow in price of values and insights. So we make our algorithm best effort and bound it to a time limit. In time limit, it starts from most similar options only to previous options and they also become diverse.

We evaluate the efficiency and usefulness of GeoGuide in an extensive set of experiments. We consider two types of experiments: first, a performance study measures the influence of relevance and size constraint thresholds on execution time. Second, we measure the usefulness of our framework in a user study.

Experiment Settings. Unless otherwise stated, we use the same settings discussed in Section **??**. All experiments are implemented in Python (functionality) and JavaScript D3 (visualization) on a 2.8GHz Intel Core i5 machine with an 16GB main memory, running OS X 10.9.2.

Performance Study. GeoGuide is designed for exploratory context where interactivity is a need. The "best-effort" greedy approach of Highlighter (Algorithm 1) guarantees to return the best possible results within a time limit. We consider a large time limit ($tlimit = 2s$) in order to evaluate the effect of relevance and size constraint thresholds on execution time.

Figure 2 left illustrates the effect of size constraint by varying k from 2 to 300. In general, larger values of σ provides more freedom for the algorithm hence more time-

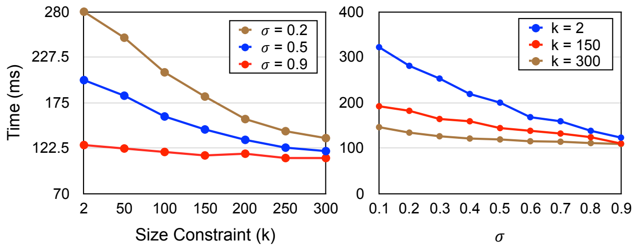

Figure 2: Performance Evaluation

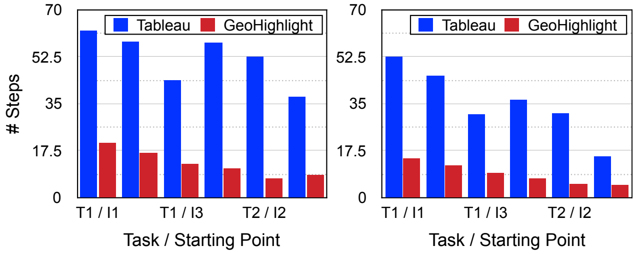

Figure 3: User Study

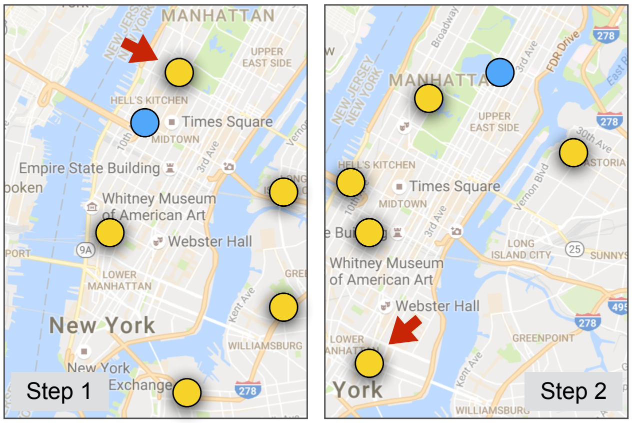

Figure 4: Application of GeoGuide

consuming. An interesting observation is that increasing k leads decreasing execution time. This is because in larger sets, there exist fewer opportunities for increasing diversity, hence Highlighter terminates early.

Figure 2 right confirms that lower values of σ decreases execution time as they provide more flexibility for diversity improvement. For larger values of k (i.e., when $k > 100$), the influence of σ on execution time becomes insignificant.

User Study. The principled question that we ask ourselves is whether GeoGuide is useful for analysts in practice. To answer this question, we designed a user study with 24 participants (students in Computer Science). Half of the participants know the New York region well (experts) and the other half have a limited knowledge (novice). In our user study, we define a task for each participant and ask him/her to fulfill the task using both GeoGuide and Tableau (as the most advanced spatiotemporal visualization tool). Then we measure the cardinality of steps to reach the goal.

We define two tasks, *T1: finding a point in a requested location*, and *T2: finding a point with a requested profile.* As an example for *T1*, we ask participants to find points in the Central Park area. An example of *T2* is to find a drop-off point with $2 tip whose trip distance is 3 kilometers. Participants may begin their navigation from three different starting points: *I1: close to the goal*, *I2: far from the goal*, and *I3: random.* We evaluate the effect of expertise, goal and starting point on the analysis length. Figure 3 illustrates the results for novice (left) and expert (right) participant.

We observe that in general, it takes in average 10.7 steps to reach a defined goal in GeoGuide, i.e., 33 steps less than Tableau. This shows that the guidance component helps analysts discover their data and quickly reach to the goal. Level of expertise improves the analysis length in average by 4 steps. Interestingly, starting points do not have a huge influence. It is potentially due to the diversity component which provides distinct options. We also observe that *T2* is an easier task than *T1*. This is potentially due to similarity component where the analyst can request options similar to what she has already seen and greedily moves to match profiles.

3. DEMONSTRATION PLAN

Demonstration attendees can participate as analysts in GeoGuide. We provide several spatial datasets in our demo session: *i*. Yelp dataset for restaurant check-ins with 229,907 points and 11,537 attribute-values, *ii*. Airbnb dataset for short-term lodging with 4,200,000 points and 2000 attribute-values, *iii*. New York taxi dataset with 173,179,759 points and 18 attribute-values, and *iv*. New York bike-sharing dataset with XXX points and XXX attribute values. We describe three demonstration scenarios as follows.

Scenario 1. We demonstrate on New York taxi dataset how GeoGuide can contribute to urban planning and fleet management. We consider an explicit goal of discovering which neighborhoods work the best for which drivers to increase the overall availability of taxis in the city. We show how a chain of cab stations can be picked by GeoGuide to increase availability.

Scenario 2. We demonstrate on Airbnb dataset how GeoGuide can contribute to approach a lodging of interest based on analyst's feedback. The attendee will observe that he/she can quickly reach to satisfying housing solutions based her previous selections. We consider the concrete case of finding a cheap lodging solution near Eiffel tower which seems infeasible.

Scenario 3. We demonstrate on Yelp dataset how GeoGuide can contribute to reach an early consensus on a restaurant. The attendee will observe that his/her preferences will be immediately captured and reflected in future highlights.

4. REFERENCES

[1] J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel. Recommendations in location-based social networks: a survey. *GeoInformatica*, 19(3):525–565, 2015.

[2] M. Boley, M. Mampaey, B. Kang, P. Tokmakov, and S. Wrobel. One click mining: Interactive local pattern discovery through implicit preference and performance learning. In *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*, pages 27–35. ACM, 2013.

[3] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel. Lars: A location-aware recommender system. In *ICDE*, pages 450–461, 2012.

[4] J. Liang and M. L. Huang. Highlighting in information visualization: A survey. In *2010 14th International Conference Information Visualisation*, July 2010.

[5] L. Lins, J. T. Klosowski, and C. Scheidegger. Nanocubes for real-time exploration of spatiotemporal datasets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2456–2465, 2013.

[6] B. Omidvar-Tehrani, S. Amer-Yahia, and A. Termier. Interactive user group analysis. In *CIKM*, pages 403–412. ACM, 2015.

[7] A. C. Robinson. Highlighting in geovisualization. *Cartography and Geographic Information Science*, 38(4):373–383, 2011.

[8] W. Willett, J. Heer, and M. Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, 2007.

[9] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *TVCG*, 22(1), 2016.