Final Year Project Final Report

SpaceKey: Exploring Patterns in Spatial Databases

**Supervisor:** Dr. Reynold C. K. Cheng

**Member:**

Yin Yue / 3035234234

Wang Jikun / 3035234715

Lukito Budiman / 3035253864
Abstract

This project is developed based on the enmonours market size of real-estate market worldwide. The demand for ideal property grows along with the market. Meanwhile, internet has been regarded as a powerful way for users to access massive property records. There are some existing applications for posting and searching for properties, like Airbnb, 28hse.com. However, existing application only supports simple filtering in terms of region, price, etc. It takes significant effort for users to locate the property with detailed demand related to surrounding elements, resulting in inefficiency in property search.

This project introduces spatial pattern matching (SPM) while it models each property demand as a spatial group keyword query. In order to fully illustrate the feasibility and the strength of this approach in practice, all the data included in this project is collected from Google Maps for spatial data and existing property search applications for property data. Apart from demonstrating the functionality of SPM, this project also serves as a data provider of all the property data available online, which are collected and combined to improve the efficiency of property search saving the user’s effort of exploring massive amount of property searching websites.

This report will be illustrating all the methodology and concept corresponding to his project. Result of the survey regarding the feasibility and value of the project will also be discussed.


**Acknowledgement**

Great appreciation to all the individuals who have provided supports during the procedure of completing this project, especially our project supervisor, Dr. Reynold C. K. Cheng.

His contributions matters to the project a lot through giving generous advice and encouragement, as well as being the coordinator of the project.
Table of Contents

Abstract

Acknowledgement

1. Introduction

2. Project Background
   1. SPM
   2. Existing solutions
   3. Motivation

3. Methodology
   1. Data Collection
      1. Pol data
      2. Property data
      3. Data storage
   2. Application of SPM
      1. Unwanted object
      2. Realistic distance
      3. Distance constraint mapping
      4. Directional relation
      5. Custom object
   3. Front-end website development
   4. Project structure

4. Experiments and results
   1. Survey

5. Conclusion

6. Future work
   1. Cooperation and Promotion
   2. Domain extension

7. Reference

8. Appendix
1. Introduction

After years’ of development, the significance of property market has come to a new stage around the world[1]. Therefore, the demand of searching for property grows along with the market expansion. In a digital world as today, searching on the Internet has become a common method for every individual. While searching for a specific property, the demand of each individual varies, and different concerns are combined for each individual, where the gap of property demand and supply searching arises.

Investigation on current market participants like Airbnb (Figure 1) and 28hse.com shows that searching with the requirement of surrounding elements is not well supported, and only simple options are supported for filtering searching results.

Figure 1. Airbnb

This project is to provide a solution to fill this gap and help users to locate ideal properties. The solution is based on spatial pattern matching (SPM) (Figure 2), which is a type of spatial group keyword query introduced in recent study. Spatial pattern is defined as a graph that contains vertices and edges with specific distance constraints. The idea is to bring the algorithm into application level to tackle the property searching problem where spatial patterns are specified by the user to compose a complex search.

The project is to develop a module that supports the functionality mentioned above, and combine it with existing applications to optimize the searching process. The module is implemented as open-source API, organized in a standalone server and work as supporting component for applications. A web-based application is implemented for demonstration of the concept.

Later part of this report will illustrate the methodology and reasoning about implementation choices in data collection, website development, and API server accordingly, followed by difficulty encountered along the way when practicing different possible methods. The survey regarding the feasibility of the project will also be discussed. Final part will be a conclusion drawn from all the discussion and future work to further improve the project.
2. Project Background

This project is developed under the significant market volume of real-estate market all around the world [1]. In the meanwhile, it has become a convenient method for users to locate properties online. After investigation into existing property searching solutions like Airbnb and 28hse.com, which are websites that enable users to explore massive amount of property records. It can be observed from these applications that they only provide user with simple filtering and categorizing. In practice, demand for property may and vary in many ways in detail. Complicated demand considering the property surroundings may arise, but it can not be fulfilled by current solutions. Therefore, an advanced searching utility is demanded.

2.1 SPM

In order to handle scenario above efficiently, this project adapts a new algorithm in research, which is categorized as spatial group keyword query, namely Spatial Pattern Matching (SPM)[2].

Spatial pattern is defined as a graph where its vertices are labelled by keywords and edges labelled with distance constraints. The idea is to bring the algorithm into application level to tackle the property searching problem where spatial patterns are specified by the user to compose a complex search.

In this project, SPM is utilized in a way to match Property. In order to optimize the user-interaction, distance constraint among PoIs are prohibited, while the distance constraints are mapped by close, medium, far, which also allows customization.

Point of interest (PoI) is the place that is related to property, correspondingly labelled with keyword, such as parking lot, supermarket, school, etc. In this project property is the target to be searched, therefore each PoI must be associated with the property in the pattern to conduct the search.
2.2 Existing solutions

In terms of property search, existing websites such as 28hse.com, Airbnb don’t provide searching utility related to PoIs, meaning that if the demand of the user is related to any PoI, the user needs to check the property surrounding manually, which will be inefficient and time-consuming for the user.

As shown (figure 3 and figure 4), existing solutions only provide simple filtering related to the property entity, which are labelled when posting corresponding records. None of them consider the surrounding PoIs related to the property, meaning that no PoI information is available for searching.
2.3 Motivation

On the contrary, this project is to provide the functionality that is missing in current solutions, by adapting SPM to fulfill the demands related to PoI associated with the target property. In order to promote the SPM as an external algorithmic support to current service providers, the website application is developed as a separated part from the SPM module.

After investigating into the property search websites, there are lots of websites available online, while providing massive amount of property data, which are relatively hard to categorize and distinguish. The concept of application is to gather all the property data available to combine the data sets and minimize the information gap between landlord and property finder and improve the overall efficiency in finding a match.

Vision of this project is to demonstrate the feasibility and power of this searching utility by implementing the website application.
3. Methodology

This section will be explaining the technical choices and reasoning in building the application and the API module, as long as some of the difficulty and limitations encountered along the development.

3.1 Data collection

As mentioned in the introduction, the purpose of the application is to simulate practical usage of a property search website, therefore both property data and PoI data are required and utilized in the searching procedure. For the SPM algorithm to operate, the coordinates of the data, including latitude and longitude, are crucial parts of the data records. Data entries without coordinates, as required by the searching procedure, are abandoned. Following subsection is to illustrate how different data are obtained through various approaches and how the data are stored in the database. The data volume obtain is around 40000 entries for PoI data, 25000 entries for property data.
3.1.1 PoI data

All PoI data in the database are collected from Google Maps. Originally, parsing from the AJAX response is attempted, meaning that obtain the PoI directly from the http response. When trying to proceed the crawling process through AJAX request and response, it is observed that Google Maps adapts a complex protocol within the searching procedure. The response returned from the searching request is a URL pointing to a txt file that contains the search result.

![Figure 5. Sample response file](https://docs.google.com/document/d/1LwIPtKA2h2XxO2ChNMG95bRn1TnLkHw_mwfHUkOaYVc/edit#

It can be concluded that the file is formatted but hard to interpret and parse the result without knowledge about the protocol, as shown in sample txt file (Figure 5).

Some of the fields in the file is filled with null which might be some information that is not provided by the corresponding location, resulting in significant difficulty of parsing the result to well-formatted data. Therefore this approach is finally abandoned and new attempts are made to collect data from Google Maps.
After abandoning the AJAX approach that is discussed above, gathering data from the HTML page generated becomes an option. The HTML content of the webpage is organized nicely (Figure 4), where the data architecture can be simply observed and the information is arranged in similar manner. By switching to this approach, the task becomes extracting elements from the HTML page. Therefore, implementation of a crawler that is able to read HTML page content is required.

The crawler depends on a Python library namely Selenium. When crawling on the websites, Selenium as a commonly used web application testing library is utilized to simulate user interactions with the browser with webdriver controlling the browser, while it can also be utilized to organize HTML page content. It enables the crawler to locate elements according to its id, class, xpath, etc. After locating the elements, the value within the HTML element can be directly retrieved.

As shown (Figure 6), there is a keyword provided from Google Maps that can be utilized as keyword label in SPM. The plus code is also provided for each PoI, which can be further translated into coordinates through pluscode.com. Along the data crawling procedure, Google Maps changed the HTML format and display contents, resulting in changes in data crawler accordingly. A map is composed by the data gathered, with each PoI classified as corresponding category. Then the map serves as the domain for SPM algorithm to conduct pattern-matching, and it is stored in the API server. The crawler run occasionally independent of the application for data update.

It’s observed that when keep the webdriver running for lots of iterations, the Google Maps will stop responding from requests, which will result in crash in crawler. Therefore, the strategy is to restart webdriver when one keyword search is completed to ensure the fluency in HTTP communication.
3.1.2 Property data

As the main concern of the application, property data is collected from 28hse.com and Airbnb, while 28hse.com provide precise location (Figure 7) and Airbnb (Figure 8) only provides approximate location for user’s privacy concern. The difference between these two data resources is that 28hse.com provides both property data for rent and sell, while Airbnb only provides data for short-term rent.

![Figure 7. 28hse.com location information](image1)

![Figure 8. Airbnb information](image2)

Similar to the approach in gathering PoI data, property data is also collected through parsing html content. While crawling data from property websites, user-interaction simulation is not required, as their protocol is simpler than Google Maps, therefore the webdriver which controls the browser can be runned in the head-less mode, saving the resources used to load UI content.
3.1.3 Data storage

Figure 9. Database table definition

As shown in the figure(Figure 9), data format is defined as database tables with null value tolerance to handle exception. While collecting spatial data and property data, it needs to be ensure that there will be no duplicate entries in the database, which is implemented by adding database constraints in each table. The uniqueness for PoI data is ensured by the combination of the PoI name and the coordinate, while the uniqueness of the property data is ensured by the redirect URL. The database is deployed on AWS RDS connected through mysql.connect, and it is for the convenience of data update and retrieval.

All the data that are collected are stored in the database, which serves as the domain for the SPM algorithm to perform the search utility, at the same time the data will be updated periodically by the crawler.
3.2 Application of SPM

This project adapts SPM to model the searching problem, but efforts were made to bring SPM into practice within the application. Subsections will be illustrating how SPM is applied to practical challenges in property search application.

3.2.1 Unwanted object

SPM is based on finding certain match among the data domain, while unwanted object is some PoI that should not be matched regardless of the distance constraint, which is logically contradicting from the nature of the algorithm.

To tackle this problem,

define the query as P, the set of unwanted object as U.

Apply SPM on P \ U, filtering the result to achieve that no unwanted object in U remains in the matching result. In other words, excluding the unwanted object when applying SPM, filter out the results in the obtained matches. This feature can be utilized when specifying the spatial pattern, which can be labelled on the PoI in the distance constraint selection.
3.2.2 Realistic distance

As originally implemented in the SPM algorithm, distance is measured in terms of euclidean distance, which is not practical enough. Two locations that are close in terms of euclidean distance may take long time to travel between. As a result, realistic distance is demanded for this project.

![Routing in Google Maps](image)

Figure 9. Routing in Google Maps

The approach is to acquire routing data through Google Maps(Figure 9), which is measured in terms of time which is expected to consume travelling between two locations. However, the volume of routing data exponentially corresponding to the volume of the PoI data volume, directly crawling the routing data is not feasible as the problem nature. As a solution, this project still utilized euclidean distance within the search algorithm with relatively looser distance constraint while obtain the routing upon getting the match result for optimization of the result display.
3.2.3 Distance constraint mapping

Along the way of utilizing SPM in a practical application manner, distance constraint is the part where the user experience can be optimized through pre-define the mapping between distance definition and numerical value, such as close and 200 meters. For not preventing the user from further customization, the user is able to decide the actual distance mapped to each distance measurement, while associate the measurement requirement with each PoI that is chosen by the user(Figure 10).

Figure 10. Distance constraint customization
3.2.4 Directional relation

There is no any directional relation settings originated from the SPM algorithm, while directional relation can be meaningful in real life scenario for some traditional concerns, namely Feng shui. By specifying directional for each PoI meaning that the PoI need to be appearing at corresponding direction in terms of the property. This functionality is achieved by applying validation on the matching result after applying the SPM to filter out the matches that are not satisfied.
3.2.5 Custom object

In real life scenario, there are some PoI that must be considered when user try to locate a property, such as work space, friend’s house, etc. The logic is to append this PoI into the domain for the SPM algorithm to achieve the match. This project enables user to explore within the map and pick custom object through click on the map (Figure 12). After specifying the location, an unique keyword will be assigned to this object.

Additionally, the custom object also supports other functionalities mentioned in section 3.2.3 and section 3.2.4. The custom object, assigned with unique keyword, is equivalent to other PoI in conducting the SPM. In other words, the custom object can also be associated with a distance constraint and a directional relation constraint within the same spatial pattern (Figure 13).
3.3 Front-end website development

For the front-end development, Semantic UI (Figure 14) is chosen as the front-end framework, which is integrated with libraries like React, Angular, etc. The framework provides various UI components which are pre-set and convenient for customization through parameter modification.

Additionally, empowered by the integration within the framework with React-JS, it becomes relatively easy to integrate with Google Maps to implement the map features, displaying labels of properties and PoIs(Figure 15). The location of the labels depends on the coordinates of the specific data entry, meaning that the location correspondence is the realistic representation of the location relationship among all the PoIs and properties. This way of representation can provide user with a clearer picture about all the property candidates and related PoIs on the map.
3.4 Project structure

![Project architecture diagram](image)

Figure 16. Project architecture

It can be seen from the architecture (Figure 16) that the API server is developed separated from the website backend to simulate the situation where the API server supports other property search application. The API server communicates with website back-end through HTTP, which maintain a protocol for further usage. Additionally, the API server is able to fetch data set from MySQL database which is hosted on the AWS RDS to update the status of the data set.
4. Experiments and Results

This section will introduce the survey and feedbacks obtained in proving the feasibility and value of this project.

4.1 Survey

![Figure 17. Importance on property selection](image)

In order to analyze the feasibility and value of the project, a survey is initiated and distributed among the potential users. The survey is about essentialness of PoIs, including what PoIs are missing in our keyword list and whether the SPM feature is crucial within a property search application.

As shown in the result (Figure 16), apart from price and location, which are basic elements for a property, neighborhood also carries an essential part in choosing an ideal property. The result indicates the potential value in search related to neighborhood, which SPM can serve as a powerful tool.

![Figure 18. Essentialness of SPM functionality](image)

As demonstrated in the question result (Figure 17) regarding whether SPM is an essential feature, where SPM is described as a feature that specifies the distance between the property and the places. The survey shows around 80% positive respond that the SPM functionality is essential, which generally proves the feasibility and value of this project.
5. Conclusion

This project developed a website application for property search, utilizing SPM as algorithmic support. Through iterations of development, the technical objectives of the project has been achieved. In applying SPM, there are improvement and optimization within the UI display and user-interaction to achieve fluent user-experience.

With the survey collecting feedback from potential customers, it proves the feasibility and the power of SPM. In terms of business value, the project still has a long way for business model development.

Overall, this project has completed the task of applying SPM in the real life scenario, and it is able to serve as a tool to improve the efficiency in property search, being the change maker in the industry.
6. Future work
This section will cover the further potential development of the project not only as a technical deliverable but also as a real-life application demo.

6.1 Cooperation and Promotion
To actually practice the business value of SPM and to ensure the data consistency, cooperate with existing property search website is necessary. This project attempt to merge and combine all the property data available in the internet and encapsulate with the SPM functionality to help user identify the ideal property. Then the user is redirected to the original page to complete the procedure.

Potential partners includes the 28hse.com and Airbnb, but not limited to these websites. At the same time, promoting the API as an external support is also the business goal of the project.

6.2 Domain extension
The project domain is currently limited to Hong Kong as the data entries obtained are all within or around Hong Kong. However, the SPM algorithm is powerful enough to utilize on any data sets, therefore adapting new data source and extend the domain of the application is an area for further development.
7. References


8. Appendix