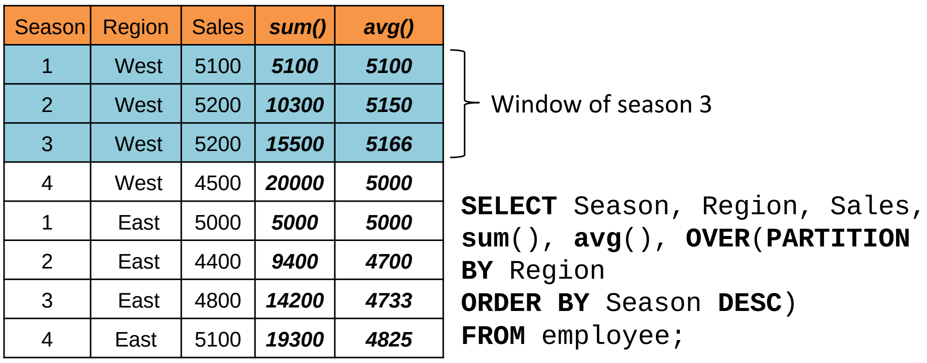
Chapter 1

Introduction

With the maturity of database technologies, nowadays applications collect data in all domains at an unprecedented scale. For example, billions of social network users and their activities are collected in the form of *graphs*; Thousand sensor reports are collected per second in the form of *time series*; Hundreds of millions of temporal-locations are collected as *trajectories*, to name just a few. Flooded by the tremendous amount of data, it is emerging to provide useful and efficient analytics for various data domains. Traditional SQL-based analytics which comprises of primitives (such as, partition, sorting and aggregation etc.) become limited in non-structured data domains. In SQL context, interesting analytics such as graph traversal and pattern detection often involve complex joins which are very hard to optimize without domain knowledge. In this thesis, we explore the neighborhood data analytic, which in SQL is expressed by the window function, on different data domains and demonstrate how to efficiently deploy neighborhood data analytic to gain useful data insights.

1.1 Neighborhood Data Analytic

By its self-describing name, neighborhood data analytic aims to provide summaries of each object over its vicinity. In contrast to the global analytics which aggregates the entire collection of data as a whole, neighborhood data analytic provides a personalized view on each object per se. Neighborhood data analytics originates from the window function defined in SQL which is illustrated in Figure 1.1.



Season	Region	Sales	*sum()*	*avg()*
1	West	5100	***5100***	***5100***
2	West	5200	***10300***	***5150***
3	West	5200	***15500***	***5166***
4	West	4500	***20000***	***5000***
1	East	5000	***5000***	***5000***
2	East	4400	***9400***	***4700***
3	East	4800	***14200***	***4733***
4	East	5100	***19300***	***4825***

Window of season 3

```
SELECT Season, Region, Sales,
sum(), avg(), OVER(PARTITION
BY Region
ORDER BY Season DESC)
FROM employee;
```

Figure 1.1: A SQL window function computing running sum and average of sales. The window of season-3 is highlighted.

As shown in the figure,the sales report of a company contains five attributes. "Season", "Region" and "Sales" are the original fields, "`sum`()" and "`avg`()" are the analytics representing the running sum and average. A window function is represented by the `over` keyword. In this context, the window of a tuple o_i contains other tuples o_j such that o_i and o_j are in the same "region" and o_j's "season" is prior to o_i's. The window of season-3 for region-"West" is highlighted.

Apart from this example, there are many other usages of the window function in the relational context. Being aware of the success of the window function, SQL 11 standard incorporates "`LEAD`" and "`LAG`" keywords which offer fine-grained specifications on a tuple's window.

Despite the usefulness, there are few works reporting the window analytics in the non-relational context. This may dues to the usage of *sorting* in relational windows. For example, in Figure 1.1, objects needs to be sorted according to "Season", and then the windows of each object are implicitly formed. However, in non-relational context, sorting may be ambiguous and undefined.

To generalize the window function to other data domains, we define the neighborhood analytics in a broader context. Given a set of objects (such as tuples in relational context or vertexes in graph context), the neighborhood analytic is a composite function ($\mathcal{F} \circ \mathcal{N}$) applied on every object. $\mathcal{N}$ is the *neighborhood function*, which contains the related objects of o; $\mathcal{F}$ is an *analytic function*, which could be aggregate, rank, pattern matching, etc. Apparently, relational window functions is a special case of such defined neighborhood analytics. For example, window function in Figure 1.1 can be represented as $\mathcal{N}(o_i) = \{o_j | o_i.season > o_j.season \wedge o_i.region = o_j.region\}$ and $\mathcal{F} = \texttt{avg}$. Since the *sorting* requirement is relaxed, our definition of neighborhood analytics enriches the semantic of relational window notations and can be applied on other domains.

1.2 Scope of the thesis

In this thesis, we explore the neighborhood analytics in different data domains. Our efforts showcase the usefulness of the neighborhood concepts in those data domains and address the efficiency issues when adopting nontrivial analytics. In particular, we looked at three most prevalent data domains, namely **graph**, **time series** and **trajectory**. We then categorize two intuitive neighborhood function $\mathcal{N}$ as follows:

Distance Neighborhood: the neighborhood is defined based on numeric distance, that is $\mathcal{N}(o) = \{o_i | \texttt{dist}(o, o_i) \leq K\}$, where `dist` is a distance function and K is a distance threshold.

Comparison Neighborhood: the neighborhood is defined based on the comparison of objects, that is $\mathcal{N}(o) = \{o_i | o.a_m \texttt{ op } o_i.a_m\}$, where a_m is an attribute of object and `op` is a binary comparator.

Due to the space limitation, we only consider the neighborhoods that is *distance* or *comparison* or any combinations of these two. In terms of analytic function $\mathcal{F}$, we consider aggregate, rank, and pattern matching.

1.3 Contributions

The high level contributions of this thesis are bi-folded. First, by sewing different $\mathcal{N}$ and $\mathcal{F}$, three interesting neighborhood analytic queries are proposed for *graph*, *trajectory* and *time series* data domains respectively. Second, this thesis deals with the efficiency issues in deploying corresponding analytic queries to handle data with nowadays scale. The overview of this thesis is as show in Figure 1.2.

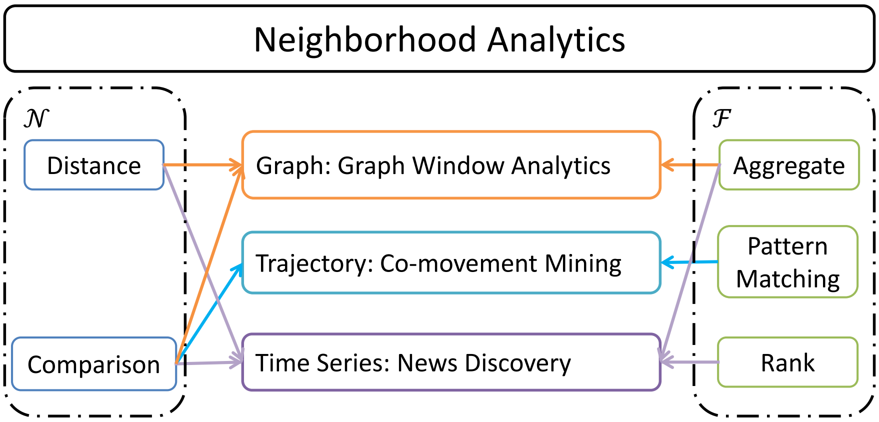


Figure 1.2: The road map of this thesis. There are three major contributions as highlighted in the center. Each contribution is a neighborhood analytic based on different $\mathcal{N}$ and $\mathcal{F}$ as indicated by arrows.

1.3.1 Graph Window Analytics

This first piece of the thesis is on proposing neighborhood analytics for graph data. Information networks such as social networks, biological networks and phone-call networks are typically modeled as graphs where the vertexes correspond to objects and the edges capture the relationships between these objects. In the context of graph, we the *graph window analytics*. The window is essentially a neighborhood function. We have identified two instances of graph windows, namely *k-hop* and *topological* windows. The *k-hop* window is a *distance* neighborhood function, i.e., $\mathcal{N}_k(v) = \{u|\texttt{dist}(v,u) \leq K\}$, which captures the vertexes that are k-hop nearby. The *topological* window, $\mathcal{N}_t(v) = \{u|u \in v.ancestor\}$, is a *comparison* neighborhood function that captures the ancestors of a vertex in a directly acyclic graph.

With the two defined neighborhood function, we present many useful analytics. To support efficient graph window query processing, we propose two different types of indexes: Dense Block Index (DBIndex) and Inheritance Index (I-Index). The DBIndex and I-Index are specially optimized to support k-hop window and topological window query processing. We develop the indexes by integrating the window aggregation sharing techniques to salvage partial work done for efficient computation. In addition, we develop space and performance efficient techniques for index construction. In our experiments, DBIndex saves upto 80% of indexing time as compared to the state-of-the-art competitor.

1.3.2 Automatic News Discovery in Time Series Data

The second piece of the thesis proposes a neighborhood analytic query to automatic discover news in the time series data. Automatic discovery of newsworthy themes from sequenced data can relieve journalists from manually poring over a large amount of data in order to discover interesting news. An typical news themes generated from time-series data is the so-called *prominent streaks.* We resolve the limitations of *prominent streaks.*

Previous efforts have focused on generating prominent streaks that are limited to single subjects. In this paper, we consider the prominence of a subject's streak in comparison to the peers' and propose a novel scoring function that takes into account both strikingness and diversity. Our objective is to maintain the k most striking and representative news themes for each subject. We study the problem in both offline and online scenarios, and propose various window- level pruning techniques to and striking candidate themes. Among those candidates, we then develop approximation methods, with theoretical bounds, to discover the k most representative themes. We conduct experiments on four real datasets, and the results demonstrate the efficiency and effectiveness of our proposed algorithms: the running time achieves up to 500 times speedup and the quality of the detected news themes is endorsed by the anonymous users from Amazon Mechanical Turk.

1.3.3 Mining Co-Movement in Trajectory Databases

The third piece of the thesis is on trajectory. Discovering co-movement patterns from large-scale trajectory databases is an important mining task and has a wide spectrum of applications. Previous studies have identified several types of interesting co-movement patterns and showcased their usefulness. In this paper, we make two key contributions to this research field. First, we propose a more general co-movement pattern to unify those defined in the past literature. Second, we propose two types of parallel and scalable frameworks and deploy them on Apache Spark. To the best of our knowledge, this is the first work to mine co-movement patterns in real life trajectory databases with hundreds of millions of points. Experiments on three real life large-scale trajectory datasets have verified the efficiency and scalability of our proposed solutions.

1.4 Thesis Organization

Chapter 2

Literature Review

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Chapter 3

Sample Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Appendix A

Sample Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.