

UNIVERSITÀ DEGLI STUDI DI PISA
DIPARTIMENTO DI INFORMATICA
DOTTORATO DI RICERCA IN INFORMATICA

PH.D. THESIS - INF/01

From Pattern Discovery to Pattern Interpretation of Semantically-Enriched Trajectory Data

Rebecca U. Ong

SUPERVISOR
Prof. Dino Pedreschi

SUPERVISOR
Mirco Nanni, Ph.D.

SUPERVISOR
Chiara Renso, Ph.D.

SUPERVISOR
Prof. Monica Wachowicz

June 1, 2011

Abstract

The widespread use of positioning technologies ranging from GSM and GPS to WiFi devices tend to produce large-scale datasets of trajectories, which represent the movement of travelling entities. Several application domains, such as recreational area management, may benefit from analysing such datasets. However, analysis results only become truly useful and meaningful for the end user when the intrinsically complex nature of the movement data in terms of context is taken into account during the knowledge discovery process. For this reason we propose a pattern interpretation framework that consists of three main steps, namely, pattern discovery, semantic annotation and pattern analysis. The framework supports the understanding of movement patterns that were extracted using some trajectory mining algorithm.

In order to demonstrate the feasibility and effectiveness of the framework, we have specifically applied it for understanding moving flock patterns in pedestrian movement. For the pattern discovery step, we have formally defined the concept of moving flock, distinguishing it from stationary flock, and developed a detection algorithm for it. A set of guidelines for setting the parameters of the algorithm is provided and a specific technique is implemented for the *radius* parameter. As for the semantic annotation step, we have proposed a guideline for selecting appropriate attributes for semantic enrichment of individual entities and of moving flocks. Two levels of annotation, which are at individual and pattern level, were also described. Finally, for the pattern interpretation step, we have combined the results obtained using hierarchical clustering and decision tree classification in order to analyse the attributes of flock members and of the flocks, and the flocks themselves.

The entire framework was tested on the Dwingelderveld National Park (DNP) dataset and the Delft dataset, both of which are pedestrian datasets based in the Netherlands. The DNP dataset contains records of observations on the movement of visitors in the park while the Delft dataset describes movement of the pedestrians in the city. As a result, some forms of interactions, such as certain groups of visitors following the most popular path in the park, were inferred. Furthermore, some flocks were linked with specific attractions of the park.

Acknowledgments

First of all, I would like to thank my supervisors (Mirco Nanni, Prof. Dino Pedreschi, Chiara Renso, and Prof. Monica Wachowicz) for guiding me throughout my Ph.D. research until the end. Thank you for the interesting and helpful discussions you have shared with me.

I am also thankful to the members of my internal committee (Prof. Giorgio Ghelli and Prof. Paolo Mogorovich) for taking the time to attend my presentations and evaluate my thesis. I am also taking this opportunity to thank the external reviewers (Prof. Harvey J. Miller and Prof. Nico Van de Weghe) for carefully reading my thesis and providing their suggestions and comments, which were helpful in improving the final version of the thesis.

I thank Prof. Pierpaolo Degano and Ilaria Fierro as well for their patience in assisting me with the bureaucratic matters.

I would also like to thank the KDD lab for the support they have provided. I will especially remember our lunches, dinners, and other out-of-the-lab activities together.

Special thanks go to my friends (special mention to Lopa, Peter, Lam, Hung, Dung, Sergiy, Naveen, Binh, Ruzhen, Sebnem, Sonia, Yun Guo, Nayam, Elena, Hedy, JIL family, Fra e Federico, Carlo M., “coinquilini/e”) who made my stay in Pisa enjoyable, and especially to those whom I shared good and bad times with.

I should thank my friends in the Philippines as well for their prayers and their words of encouragement. Thank you also for continuing to keep in touch despite the distance.

I am grateful to my relatives, especially to my Uncles and Aunts. I would not have made it here without your help.

I would like to dedicate this thesis to my parents and my brothers as a sign of my appreciation of their love and support.

Finally, I thank God for giving these wonderful people in my life, and for allowing me to experience the Ph.D. life in Pisa.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Problem Statement	4
1.3	Contribution	6
1.4	Organization of the Thesis	9
2	Knowledge Discovery in Databases for Movement Data	11
2.1	Knowledge Discovery in Databases and Geographic Knowledge Discovery	11
2.1.1	Knowledge Discovery in Databases	11
2.1.2	Geographic Knowledge Discovery	13
2.2	The Input	14
2.2.1	Movement Data	14
2.2.2	Trajectories	14
2.2.3	Complexity of Movement Data	15
2.3	Movement Pattern	16
2.4	Semantic Annotation of Trajectories	20
2.4.1	Semantics	21
2.4.2	Semantic Annotation of Trajectories	21
2.5	Data Mining	24
2.5.1	Classical Data Mining Algorithms	25
2.5.2	Mobility Data Mining	28
2.6	Pattern Interpretation	35
2.6.1	Visual Analytics Tools	35
2.6.2	Ontology-based Systems	36
2.7	Summary and Conclusions	38
3	Methodology	41
3.1	Pattern Discovery Phase	42
3.1.1	Moving Flock Definition	42
3.1.2	The Moving Flock Discovery Algorithm	44
3.1.3	Selection of the <i>Radius</i> Parameter	49
3.1.4	Validation of the Flock Discovery Algorithm	51
3.2	Semantic Annotation Phase	53
3.2.1	Semantic Attribute Source	54
3.2.2	A Guideline for Semantic Attribute Selection	58
3.2.3	Two Levels of Semantic Annotation	61

3.3	Pattern Analysis Phase	62
3.3.1	Hierarchical Clustering for Pattern Analysis	62
3.3.2	Classification for Pattern	67
3.4	Summary of Discussion and Conclusions	68
4	Experiments	71
4.1	Datasets	71
4.2	Moving Flock Discovery	73
4.2.1	Moving Flock Results	73
4.2.2	Selection of the <i>Radius</i> Parameter	78
4.2.3	Effect of Varying the Radius Value	81
4.2.4	Effect of Ordering of Entities	82
4.2.5	Validation of the Moving Flock Algorithm	83
4.2.6	Summary of Results	85
4.3	Semantic Annotation	87
4.3.1	DNP	87
4.3.2	Delft	90
4.4	Pattern Analysis	91
4.4.1	DNP	92
4.4.2	Delft	99
4.5	Overall Summary	104
5	Conclusions	107
5.1	Contributions	107
5.2	Future Works	110
	References	113

List of Figures

1.1	Knowledge about the movement behavior of prospective customers can aid in making decisions about business ventures.	2
1.2	Analysis of a Flock Pattern.	2
1.3	A specific instance of flock pattern that can be interpreted in several possible ways depending on the movement context.	3
1.4	The trajectories in the DNP, Fontainebleau and Delft datasets.	9
2.1	Overview of the steps constituting the KDD process. (Based on [29])	12
2.2	Movement Data of an Object	14
2.3	Classification of Movement Data [25]	17
2.4	Convergence pattern. [70]	17
2.5	e_j is in front of e_i . [4]	20
2.6	An example of a mined T-pattern.	30
2.7	Proposed framework for semantic trajectory knowledge discovery. [3]	35
2.8	The trajectory semantic enrichment process. [11]	37
2.9	Overview of the Athena system architecture. [11]	37
2.10	Analysis of Commuter destinations. [84]	38
3.1	Overview of the steps constituting the KDD process. (Based on [29])	41
3.2	Proposed framework for interpretation of movement patterns.	42
3.3	Sampling of points at regular time interval.	45
3.4	Computation of the base trajectory's neighbors at each time step.	46
3.5	Order of merging candidate flocks at each time step into flocks that persist over a certain time duration.	47
3.6	Example of computing the flock extent for a flock consisting of two members.	48
3.7	Sample Plot of k -th Distances.	50
3.8	The left plot is a moving flock example while the right plot is a non-example.	51
3.9	A specific instance of flock pattern that can be interpreted in several possible ways depending on the movement context.	53
3.10	Users' responses to their interest in museum planning. [63]	54
3.11	Users' responses to their profession/position. [63]	55
3.12	A world climate map. [93]	56
3.13	A topographic map of Georgia. [93]	56
3.14	Some POIs in Pisa.	57
3.15	Fragment of 2006 Census Data for Central Northern Sydney. [9]	58

4.1	(a) The entire trajectories of moving flock members 139, 141 and 140 in the DNP dataset. (b) The trajectory segments belonging to the moving flock whose members include 139, 141 and 140 in the DNP dataset.	74
4.2	The base trajectories of the moving flock patterns found in the semi-synthetic version of the DNP dataset when the radius is set to 150m using (a) the whole trajectory dataset and (b) a Google map as background.	75
4.3	The number of moving flocks versus the number of stationary flocks in the semi-synthetic version of the DNP dataset.	76
4.4	Flock 0 and Flock 1 discovered from the Fontainebleau dataset using a radius of 150m.	76
4.5	The base trajectories of the moving flock patterns found in the Fontainebleau dataset when the radius is set to 150m using (a) the whole trajectory dataset and (b) a Google map as background.	77
4.6	The number of moving flocks versus the number of stationary flocks in the Fontainebleau dataset.	77
4.7	The base trajectories of the moving flock patterns found in the Delft dataset when the radius is set to 50m using (a) the whole trajectory dataset and (b) a Google map as background.	78
4.8	The highest-ranking (left) and the lowest-ranking (right) moving flocks discovered in the Delft dataset using 50m as the radius and 60s as the synchronization rate.	79
4.9	The number of moving flocks versus the number of stationary flocks in the Delft dataset.	79
4.10	Plot of k -th Distances for the Semi-synthetic DNP Dataset.	80
4.11	Plot of k -th Distances for the Fontainebleau Dataset.	80
4.12	Plot of k -th Distances for the Delft Dataset.	81
4.13	Semantic Annotation of Individual Flock Members in DNP.	88
4.14	Semantic Annotation of Discovered Flocks in DNP.	90
4.15	Semantic Annotation of Individual Flock Members in Delft.	91
4.16	Semantic Annotation of Discovered Flocks in Delft.	92
4.17	Sample relations among individual attributes of flock members in semi-synthetic DNP.	94
4.18	Sample relations among flock properties in semi-synthetic DNP.	95
4.19	Relations among the flock patterns found in the analysis step.	96
4.20	The trajectories of <i>FLOCK_8</i> and <i>FLOCK_9</i>	96
4.21	The trajectories of <i>FLOCK_0</i> and <i>FLOCK_1</i>	97
4.22	Decision tree obtained based on individual attributes of flock members when the target class is <i>Flock0</i> and <i>Flock2</i> , respectively.	97
4.23	Decision tree obtained based on individual attributes of flock members when the target class is <i>Flock9</i>	98
4.24	Decision tree obtained based on flock attributes when the target class is <i>main_activity_1</i>	98
4.25	Sample relations among individual attributes of flock members in Delft. . .	100
4.26	Trajectories belonging to <i>flock1</i> with OpenStreetMap as background. . . .	100
4.27	Sample relations among flock attributes in Delft.	101
4.28	Relations among the flock patterns found in the analysis step.	102
4.29	Subset of flock similarities in the un-weighted hierarchical clustering result. .	102

4.30	Decision tree obtained based on individual attributes of flock members when the target class is <i>Flock1</i>	103
4.31	Decision tree obtained based on flock attributes when the target class is <i>group_1</i>	103
4.32	Decision tree obtained based on flock attributes when the target class is <i>group_4</i>	104

List of Tables

2.1	Comparison of flock discovery approaches using the taxonomy provided by Wood and Galton [92].	33
3.1	The given pair of flocks to be compared.	65
3.2	Result of performing straightforward matching.	65
3.3	The result after performing diagonal matching.	66
3.4	Obtaining the final similarity score between the given pair of flocks.	66
4.1	Description of datasets used for the experiments.	72
4.2	Discovered moving flock patterns in DNP.	74
4.3	Top three moving flock patterns in the semi-synthetic version of the DNP Dataset.	75
4.4	Bottom three moving flock patterns in the semi-synthetic version of the DNP Dataset.	75
4.5	Discovered moving flock patterns in the National Fontainebleau Forest Park.	76
4.6	Discovered moving flock patterns in the Delft Dataset when radius is set to 40m and 50m.	78
4.7	Discovered moving flock patterns in a subset of the semi-synthetic DNP Dataset when the radius is set to varying values.	82
4.8	The set of parameter values used for the datasets.	82
4.9	Difference between the original datasets and their corresponding reordered versions.	82
4.10	Difference between the results obtained from the original datasets and their corresponding reordered versions.	83
4.11	Moving flock results for different randomized versions of the semi-synthetic DNP dataset.	84
4.12	Two moving flocks obtained from two randomized versions of Delft.	84
4.13	Moving flock results for different randomized versions of a subset of the semi-synthetic DNP dataset.	85
4.14	Mapping of semantic attributes to Wood and Galton's criteria.	88

Chapter 1

Introduction

This chapter introduces the general idea of the thesis, which is to support the task of interpreting movement patterns for the purpose of understanding movement behaviors. The discussion starts with the motivation for the thesis before stating the problem statement that it addresses. After which, contributions of the thesis are discussed. Finally, the chapter closes with an overview of the discussions that can be expected in the subsequent chapters.

1.1 Motivation

Large collections of data on movement of objects are becoming more and more accessible due to advances in mobile and location technologies such as GPS, GSM, UMTS, Bluetooth, Wi-Fi, Wi-Max, and RFID. With the use of proper tools and techniques, a vast wealth of information can be extracted from these collections. This information is generally useful in understanding the environment in which the objects move, and the movement behavior of a specific group of individuals, such as the customers of a supermarket or the motorists in a city. Understanding these behaviors can lead to better management of traffic in a city, help in choosing a good location for opening a new business or a new branch, help in selecting the types of attractions that should be further developed in a recreational area, and provide support for many other applications. Figure 1.1 provides an example of how understanding movement behavior can be useful in the business domain. Specifically, an entrepreneur who has the knowledge that a considerable number of young adults are fond of visiting a specific group of monuments in the zone (Musée Rodin, Musée d'Orsay, and Palais de la Légion d'Honneur in Paris) altogether, along with other business considerations such as feasibility studies, can materialize a business proposal and make good decisions about the venture.

In order to obtain knowledge about the movement behavior of a target group, movement data (i.e., data describing the movement of entities within a specific area) must be subjected to the Knowledge Discovery in Databases (KDD) process. This process basically involves preprocessing the input data, extracting useful information in the form of patterns from the preprocessed data, and postprocessing and analyzing the obtained patterns before finally obtaining the desired knowledge that is meaningful and interesting for the end-user.

Though most of the works in KDD literature focus on the extraction of patterns (i.e.,

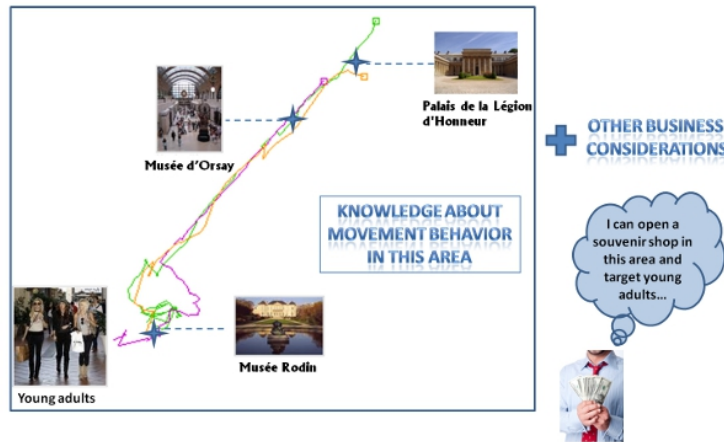


Figure 1.1: Knowledge about the movement behavior of prospective customers can aid in making decisions about business ventures.

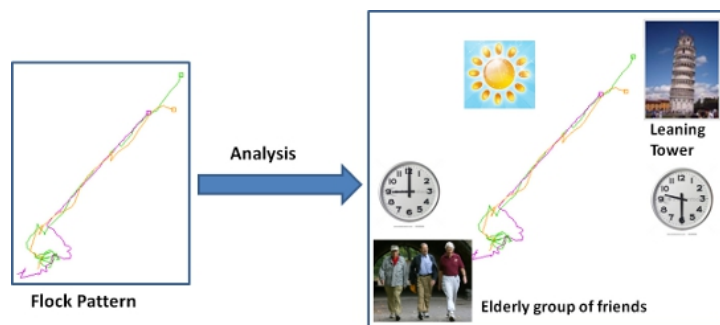


Figure 1.2: Analysis of a Flock Pattern.

the data mining phase), the postprocessing and analysis part (i.e., the interpretation phase) of the extracted patterns is important as well since the patterns by themselves are simply representations of the desired knowledge and require further processing to facilitate knowledge delivery to the users. This is illustrated in Figure 1.2. The left part of the figure is an example of a flock pattern, which basically represents a group of moving entities that are spatially close together for a certain time interval. Note that this pattern by itself may provide the user with some basic information such as the start and end time of flocking, and the IDs of the flocking entities. However, several other information describing the context (i.e., the set of facts and circumstances surrounding the situation in which the movement occurred) are still missing. For this reason, analysis is usually performed with reference to some geographical information, which adds semantics to movement data. Aside from geographical information, other possible examples are the weather condition during the time of movement and the characteristics of the moving entity, such as age or occupation. We refer to semantics as the set of concepts that are used to annotate the data for the purpose of integrating a description of the context into it. Going back to Figure 1.2, analysis of the flock pattern through the incorporation of a set of semantic information allows the user to understand that the flocking occurs among a group of elderly friends who are heading towards the leaning tower on a sunny morning. Such analysis is

important for recreational area managers, for instance, since the obtained interpretation sheds light to the meaning of extracted patterns and in turn, increases the understanding of the movement behavior of pedestrians and their impact in the managed area.

Aside from the issue of having only a small number of research efforts that focus on the interpretation phase of the KDD process, most of these works only concentrate on geographic information to understand the context of the extracted patterns. It is important to consider other forms of semantics for a fuller coverage of the movement context.

Taking the movement context into consideration is crucial since it can greatly influence the interpretation of movement patterns. To further emphasize this point, consider the flock pattern illustrated in Figure 1.3, which demonstrates different interpretations of the same pattern illustrated in Figure 1.2. It could be that the pedestrians are moving together due to interactions such as walking forward at a given speed. In this case, pedestrians tend to calculate the trajectory of their current target and turn to face the direction of the target, hence, tending to move towards the shortest path from their local origin to their destination by simply following certain paths (c) or heading for a similar area of the landscape (a). On the other hand, the flock patterns may emerge from social interactions in which the relation among the group of entities may also cause them to flock together as seen with the example of a group of friends moving around a shopping area (e) or maybe sharing a common interest in mountain hiking (b). Still another interpretation is that the flock members consist of elderly folks who tend to choose a certain route for convenience purposes (f). The interpretation of the pattern varies depending on the movement context. Thus, it is impossible to obtain a meaningful interpretation without it.

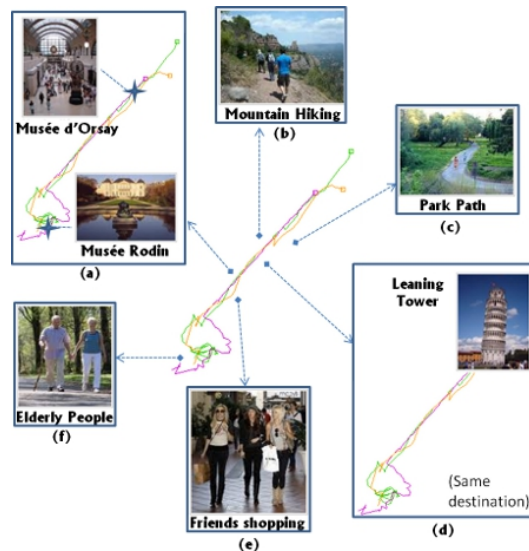


Figure 1.3: A specific instance of flock pattern that can be interpreted in several possible ways depending on the movement context.

This study addresses the issue of considering the movement context by proposing a framework that covers both the data mining and interpretation phases of the KDD process. At the same time, the framework supports pattern interpretation by using explicit thematic attributes that are available in the dataset to semantically annotate extracted patterns,

contrary to most of the current practices in data mining wherein such attributes are usually ignored for efficiency purposes.

We have applied the framework to interpreting a specific type of pattern, called flock, which was shown in previous examples. Analysis of such patterns is interesting since they represent the movement behavior of collectives rather than individuals, with the study of collective behaviors being more scalable and more useful for area managers. In addition, a flocking behavior also includes the interactions that occurred among the flocking entities and their environment. Considering these interactions allows a deeper understanding of the collective's behavior and their impact in the managed area. This is valuable in application domains such as recreational area management, traffic management and animal monitoring domains. In the recreational area management, for instance, understanding the flocking behavior of pedestrians can help in making decisions about path widening, about adding more attractions and the type of attractions that should be added, and about relocation of certain attractions.

1.2 Problem Statement

The main problem addressed in the thesis revolves around the following question:

How can semantics be used to support understanding movement behaviors represented by movement patterns?

It is well-accepted that semantics plays an important role in the KDD process due to the fact that such information allows the delivery of *meaningful* results (i.e., knowledge that is useful and interesting). This is specially true in the Web domain and this has led to the birth of Semantic Web, which has been well-studied and quite established as an active research area. For the processing of movement data in the geographic domain, however, the integration of semantics into the KDD process is quite new and research interest has only started to grow in the most recent years. For this reason, we perceive research in understanding movement behavior through semantic-enrichment of movement data as a fresh, interesting and worthwhile undertaking.

The main goal of the study is to aid the user in understanding discovered patterns. We achieve this goal by addressing the following objectives:

1. Provide an approach for the semantic enrichment of movement data and patterns
2. Provide an approach for the analysis of semantically-enriched movement data and patterns
3. Demonstrate the validity of the framework by applying it to real-world datasets

The succeeding paragraphs provide a discussion of the issues involved in each of the previously enumerated objectives.

Objective 1: Semantic Enrichment of Movement Data and Patterns Though semantic annotation for the Web has been a well studied topic as demonstrated in surveys provided in [86, 78], the set of semantic annotation tools for movement data is quite new and hence, not as rich. As a consequence several issues are yet to be addressed and

resolved in relation to the semantic enrichment of movement data and patterns. Two of these important issues are reflected by the following questions:

1. At which level should the movement data and patterns be annotated?
2. Which of the available semantic attributes should be incorporated into the movement data and patterns?

The first question addresses the need for specifying the level at which the data and patterns are annotated. This level is crucial since this can influence the depth of the interpretation results that can be obtained from the semantically-enriched data. Analysis of individually annotated entities can give way to meaningful interpretations of individual behaviors but this is not scalable when there is a large number of individuals to consider in the dataset. Furthermore, this is probably not very useful for most end-users, especially area managers, since they are usually concerned with the overall behavior of the individuals moving in the managed areas. On the other hand, analysis of semantically annotated data at an aggregate level, can provide meaningful interpretations related to the overall behavior of the moving entities. However, caution should be taken at this level in order to avoid producing annotations that are too abstract and hence, becoming meaningless. A balance between individual and aggregate level annotations is necessary.

Meanwhile, the second question addresses the need for a guideline on the selection of relevant attributes for semantic annotation. This is helpful in minimizing the size of the data that would be annotated and processed further for interpretation purposes. Since semantics of movement depend on the application domain, it is difficult to find a general and standard model for semantically-enriched movement data that can be used in any application domain. For example, mode of transportation is an important semantic attribute of movement in the context of a traffic management system but this is not as important in the context of monitoring fish behaviors. There are infinitely many possible semantic attributes if we consider all possible application domains. Thus, there is a need to analyze which semantic attributes are common and sufficient enough for describing movement behavior in the considered domain.

Objective 2: Analysis of Semantically-Enriched Data and Patterns Aside from the need for semantic enrichment of movement data and patterns, there is also a need to provide an approach for analyzing this enriched data in order to infer meanings from the patterns. A specific technique for extracting meanings from the enriched patterns should be provided to help analysts in interpreting patterns generated by data mining algorithms. This technique should answer the following questions:

1. How can the semantically-enriched movement data and pattern be transformed into meaningful patterns?
2. What type of interpretations can be inferred using this technique?

The first question addresses the important issue of designing and developing an approach for inferring meaningful interpretations from semantically-enriched data. On the other hand, the second question recognizes the capabilities and the limitations of the proposed approach.

Objective 3: Application to Real-World Datasets After designing and developing a proposed solution that achieves the previous objectives, it is also necessary to evaluate the solution’s feasibility and effectiveness. Since we are dealing with semantics, which depends on real-world context, the solution must be tested on real-world datasets. Successful application of the solution to such datasets confirms its feasibility while the meanings obtained using the proposed solution can validate its effectiveness.

In our experiments, we have instantiated and applied the framework to interpreting moving flock patterns, which are group of entities that remain spatially close together while moving from one location to another during a specific time duration, in the context of pedestrian movement. In relation to this, there are two important issues worth noting:

1. Existing flock algorithms do not perform the additional step of distinguishing between moving and stationary flocks.
2. Few research efforts were devoted to understanding the flocking behavior of pedestrians.

Concerning the first issue, we consider it important to distinguish between moving flocks (i.e., entities that remain close together while moving from one location to another) and stationary flocks (i.e., entities that remain close together while staying only in one location) because these can be considered as 2 different types of patterns, the latter being similar to meet patterns (i.e., entities that stay together in one area). Thus, they have different semantics. For instance, stationary flocks can help in understanding the specific locations wherein the individuals tend to converge. Meanwhile, moving flocks can support the understanding of how collectives as a whole move from one interesting location to another. The latter is usually more interesting in the context of tourism management, for example, since the main concern is to manage the attractions based on the movement behavior of collectives rather than individuals. Moreover, understanding how group of visitors move in a specific area provides more information compared to understanding the specific attractions that individuals are interested in.

As for the second issue, research efforts on flocking behavior have been mostly associated with collective movement of a large group of birds, fish, insects, and certain mammals as seen in [57] for theoretical ecology. However, only a few studies were focused on pedestrians, most of which concentrate on the simulation of human behavior in panic and evacuation situations [80, 26, 42, 96], and in dispersion and epidemic studies[16, 21]. In fact, no research effort has been found in studying flock patterns in tourism management despite the fact that it could enhance the management of a destination in terms of improving the access to attractions, the visitor expenditure within regions, as well as improving marketing strategies in destinations. Through the application of the proposed framework on interpreting flocking patterns of pedestrians, our work provides a contribution towards understanding flocking behavior among pedestrians.

1.3 Contribution

The main contribution of this thesis is the formulation of a framework for pattern interpretation, which addresses the three objectives enumerated in the preceding section. The framework includes three steps, which are pattern discovery, semantic annotation,

and pattern analysis. The pattern discovery step utilizes existing data mining algorithms in order to discover and extract patterns from the input movement data. The next step of semantic annotation, which addresses the first objective (i.e., semantic enrichment of movement data and patterns), involves exploiting semantic attributes that are explicitly available in the dataset in order to take the movement context into account. Finally, the pattern analysis step, which addresses the second objective (i.e., analysis of semantically-enriched data and patterns), involves the application of data mining techniques on the enriched data and pattern in order to infer meaning from them. Thus, the framework covers the KDD process more fully compared to existing approaches since it does not only include either the data mining or the interpretation phase of the KDD process. Instead, it encompasses both phases with the pattern discovery step corresponding to the data mining phase of KDD, and the semantic annotation and the pattern analysis steps corresponding to the interpretation phase. The last objective (i.e., application to real-world datasets) is achieved by instantiating the framework to the interpretation of moving flocks, which is an important type of movement pattern due to their capability to represent collective movement behaviors as well as the interactions among their members. A set of tools that support the three steps of the framework for moving flock patterns were implemented. Moreover, the framework was tested on different pedestrian datasets, which will be described in a later part of this chapter.

The novelty of our approach with respect to other approaches in analyzing movement data is that we explicitly consider the thematic attributes of moving individuals, such as the age and the occupation of park visitors, and mined patterns in order to find correlations among them. This is important since it is impossible to infer the obtained interpretation results (refer to Chapter 4) without taking the thematic attributes into account. The framework initiates a first step towards the explanation of why the mined patterns occurred, how the patterns are related to each other, and which are the semantic aspects that make the patterns correlated.

During the realization of the proposed framework, we have also achieved different contributions corresponding to the different steps of the framework. These are summarized in the next three paragraphs.

Pattern Discovery Step For the pattern discovery step, we have introduced the notion of moving flocks and have also provided a formal definition that distinguishes them from stationary flocks. Compared to existing works on flock discovery, which do not make this distinction, we chose to differentiate between the two since these correspond to different patterns and thus, have different semantics. In addition to defining moving flocks, we have also developed and implemented a moving flock discovery algorithm for extracting such patterns.

Semantic Annotation Step Two important issues related to semantic annotation of movement data that were mentioned in the previous section are the following: (1) at which level should the movement data be annotated?, and (2) which of the available semantic attributes should be incorporated into the movement data and the discovered patterns? We address the first issue by proposing two levels of semantic annotation, namely individual and pattern level. Annotating at these two levels allow the interpretation of the different facets of a pattern. On the other hand, we address the second issue by providing a guideline

for selecting semantic attributes based on the criteria provided by Wood and Galton in [92]. While these criteria were used to classify collectives in [92], we used the same criteria to filter out unnecessary data for annotation.

Pattern Analysis Step Finally, for the pattern analysis step, we propose a combination of executing hierarchical clustering and decision tree induction classification algorithms applied to individual and flock properties, and to flock instances themselves in order to support the discovery of meaningful interpretations from the flock patterns. While existing data mining techniques were used for the pattern analysis step, the idea of applying them on attributes (instead of dataset entries) for the purpose of pattern interpretation is novel.

Datasets The framework was tested on two different pedestrian datasets, which are the Dwingelderveld National Park (DNP) and the Delft dataset, to demonstrate the applicability and effectiveness of the framework on different contexts.

The DNP dataset contains data about visitors' movement in a Dutch recreational park, which consists of short and long trails for walking, cycling and horseriding. It is a very popular area that receives between 1.5 and 2 million visitors per year [87]. It also includes several attractions such as bird watching lookouts, sheep farms, a teahouse, and heath lands.

On the other hand, the Delft dataset is situated in a Dutch city named Delft. It is known for its typically Dutch town center and its canals. Some of the city's notable and historical buildings are Oude Kerk (Old Church), Nieuwe Kerk (New Church), the Prinsenhof (Princes' Court), its City Hall, the Oostpoort (Eastern gate), the Gemeenlandshuis Delfland, and Waag (Weighhouse) [23]. It also houses the Delft University of Technology, which is one of three universities of technology in the Netherlands. Its attractions, in general, include churches, museums, factories, windmills, botanical gardens, markets, restaurants, and shopping areas. It is worth noting that the two datasets are set in different contexts based on the given descriptions of the associated area.

Furthermore, the first step of the framework was tested on the Fontainebleau dataset. While the setting of DNP and Delft are in the Netherlands, Fontainebleau is located in France. Like DNP, Fontainebleau is also a recreational park. More specifically, it is a massive wooded area of 25,000 ha, 21,600 ha of which are currently supervised by a national park management body. Its wild landscape attracts a considerable number of hikers, rock-climbing or mountain-biking enthusiasts, horse riders, cyclists and Sunday walkers [30]. In fact, millions of visitors come to the park every year (e.g., 13 million in 2006). The routes are usually used for walking and they can probably be dated back to the sixteenth century. Its forest consists of wild plants and trees, and a population of birds, butterflies and mammals.

The DNP dataset contains a total of 141,826 sample points of 372 visitors whose tracks were recorded using GPS devices given to them at the parking lots where their visits started. This data collection was carried out once a month during spring and summer of 2006, having in total 7 days (weekend and weekdays) of tracking. After preprocessing, the dataset consists of 370 trajectories as seen in the first image shown in Figure 1.4.

The Fontainebleau dataset, on the other hand, contains 22,748 sample points of 23 visitors. Their movements were tracked using GPS devices as well. These data were collected during the months of April, May, and September, having a total of five days in

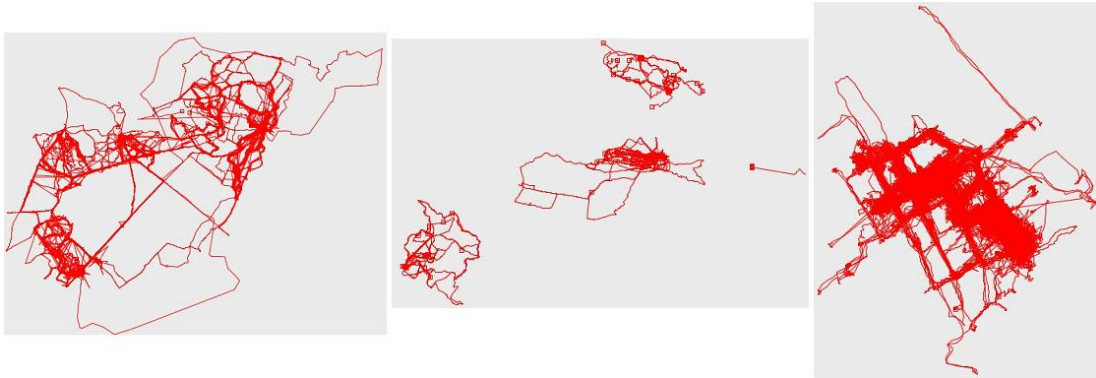


Figure 1.4: The trajectories in the DNP, Fontainebleau and Delft datasets.

the year 2004. The dataset contains a total of 207 trajectories, which is illustrated in the second image included in Figure 1.4, after preprocessing. This implies that some visitors have more than one trajectories, which represents the different trips that a visitor has made.

Finally, the Delft dataset consists of 467,454 sample points, which are collected from 285 pedestrians from the 18th to the 21st of November in the year 2009. After preprocessing, the dataset contained 303 trajectories, which are shown in the last image found in Figure 1.4.

Through the application of the framework to the DNP and Delft datasets, we were able to infer meaningful interpretations, such as the tendency of visitors to flock together in the most popular route of DNP or the tendency of the visitors to flock due to sharing common interests. This demonstrates the feasibility and the effectiveness of the framework. It can also be applied to other real-world datasets and help end-users, especially recreational area managers, in understanding movement behaviors of visitors and consequently, making decisions that can lead to improvements in the managed area.

1.4 Organization of the Thesis

The succeeding chapters are organized as follow:

Chapter 2 includes a discussion of preliminary concepts used in the thesis, and a discussion of related works. It covers the explanation of fundamental concepts such as the KDD process, movement data, trajectories, and movement patterns. It also provides an overview of existing works on general data mining tasks with a special focus on flock discovery algorithms, and on pattern interpretation systems.

Then, the discussion of the proposed pattern interpretation framework is covered in Chapter 3. It is basically split into three parts corresponding to the three steps of the framework. Meanwhile the application of the framework to real-world dataset and the conducted experiments are elaborated in Chapter 4. The discussions found in these chapters are extensions of those found in the following publications:

M. Wachowicz, R. Ong, C. Renso, and M. Nanni
Discovering Moving Flock Patterns among Pedestrians through Collective Coherence

CNR-ISTI Technical Report 2010-TR-027

To appear in the International Journal of Geographical Information Science, 2011

R. Ong, M. Wachowicz, M. Nanni, and C. Renso

From Pattern Discovery to Pattern Interpretation in Movement Data

Accepted paper in the Third International Workshop on Semantic Aspects in Data Mining (SADM 2010)

Workshop Date: 14 December 2010

In conjunction with the 2010 IEEE International Conference on Data Mining

In IEEE ICDM Workshop Proceedings, pp. 527-534

Lastly, the conclusions obtained from the study as well as new directions for future works are found in Chapter 5.

Chapter 2

Knowledge Discovery in Databases for Movement Data

This chapter provides a discussion of the state of the art with respect to our proposed framework for pattern interpretation. It starts with the big picture by providing an overview of the KDD (Knowledge Discovery in Databases) process and a specialized area of KDD for geographic data, called GKD (Geographic Knowledge Discovery).

Afterwards, the bits and pieces of the overall KDD process for handling movement data will be discussed based on the sequence of KDD steps, which is shown in Figure 2.1. The discussion will start with preliminary concepts relating to the input and output of the process before dealing with the sequence of steps necessary for transforming the input movement data into the desired output.

2.1 Knowledge Discovery in Databases and Geographic Knowledge Discovery

This section provides an overall picture of Knowledge Discovery in Databases and its specialized field known as Geographic Knowledge Discovery.

2.1.1 Knowledge Discovery in Databases

The KDD (Knowledge Discovery in Databases) process deals with a large collection of data, which contains a hidden wealth of information or knowledge that can be discovered through the proper application of the process. KDD is described as “the overall process of converting raw data into useful information” in [29], while a consistent definition in [83] describes KDD as the “overall process of discovering useful knowledge from data”. It has several applications in different fields, such as business, science, and engineering. In the business domain, for instance, point-of-sale data that describe the transactions performed by customers can now be collected. Proper processing of such data can support business applications like customer profiling and targeted marketing. Another example is in the field of ecology in which data about the movement of animals can now be gathered. Using this data as input to the KDD process, information such as the usual route taken by a large percentage of the animals during migration periods, or the locations wherein

the animals tend to converge can be obtained. This information, in turn, can support the understanding of how the observed animals interact with each other and their environment.

KDD consists of a sequence of transformation steps as shown in Figure 2.1. The first three steps (including selection, preprocessing, and transformation) can be combined into a larger step called preprocessing. Hence, KDD can be described as comprising of three main phases, which are preprocessing, data mining and interpretation/evaluation.

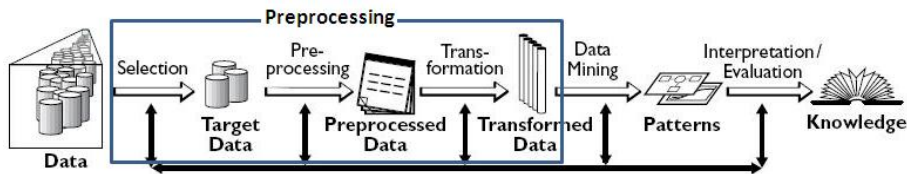


Figure 2.1: Overview of the steps constituting the KDD process. (Based on [29])

An overview of these phases is discussed in the succeeding paragraphs but a more in-depth discussion for application to movement data will be provided later in this chapter.

The Preprocessing Phase

The preprocessing phase is mainly concerned with cleaning the input data and transforming it to a format that allows it to be subjected to the next phase, which is data mining. As shown in Figure 2.1, it can further be subdivided into three smaller steps and these include selection, preprocessing, and transformation.

The Selection Step Selection involves choosing a subset of the data from which knowledge will be mined. This is applicable in cases wherein the end-user is only interested in the data observations that occurred at a certain time period, those that were collected from a specific area, or other similar types of constraints. Aside from selecting the data that the user is interested in, the attributes or features of the data are also filtered to retain those that are relevant for obtaining the desired analysis result.

The Preprocessing Step Preprocessing involves cleaning the data to remove noise observations, which can include duplicate entries, or entries with dubious values for some attributes. For example, if the age attribute in the dataset was restricted to range from 18-40, entries with values falling outside of this range should be cleaned either by removing these entries or by correcting the spurious values.

The Transformation Step Lastly, transformation ensures that the input data is converted in an appropriate format for the subsequent data mining step. For example, some mining algorithms for movement data require that spatial points must be encoded in the longitude/latitude format.

After the completion of these three steps, the final output of the preprocessing phase is referred to as the transformed data.

The Data Mining Phase

The second main phase of KDD is data mining. It entails the use of a sophisticated mining algorithm that extracts regularities, anomalies or other interesting relations from the transformed data and outputs them in the form of patterns. Classification, clustering, and association mining algorithms are standard types of data mining algorithms. Aside from these, the set of data mining algorithms has been continually growing in number and in sophistication due to the heterogeneity of data types to deal with and the variety of analysis tasks to be performed on them.

The Interpretation/Evaluation Phase

The patterns mined in the previous step are processed further during the interpretation/evaluation phase for the purpose of extracting meaningful and useful information, which is referred to as *knowledge* in Figure 2.1. It involves filtering invalid and useless patterns. Moreover, analysis steps are performed on the remaining patterns in order to infer meaningful interpretations. Upon the completion of the process, the desired analysis result should have been communicated to the end-user.

It is also important to note that in the execution of the KDD process, it is possible to revert and repeat certain sequence of steps, as indicated by the shaded arrows in the figure until the desired information is obtained.

2.1.2 Geographic Knowledge Discovery

Since the focus of the thesis is KDD in the movement context, we provide a separate discussion of GKD (Geographic Knowledge Discovery) in this subsection. GKD, which is a term first introduced by Miller and Han in [60], is a specialized field of KDD that focuses on data related to geographic space or location.

The GKD process in the context of movement data can be described as the overall process of converting movement data into useful knowledge that can support decision making in geographic-related applications. It can be seen as consisting of three phases, which are analogous to the main phases of KDD. These include trajectory reconstruction, knowledge extraction and knowledge delivery as described in [35].

Trajectory reconstruction, which corresponds to the preprocessing phase of KDD, is concerned with converting raw movement data (i.e., movement data that has not been preprocessed) to trajectories of individual moving objects, and the storage of these resulting trajectories. Once the trajectories are reconstructed and stored, useful patterns are extracted during the knowledge extraction phase by applying spatio-temporal data mining methods. This phase corresponds to the data mining step of KDD. Finally, the last phase called knowledge delivery involves reasoning, interpreting and presenting extracted patterns to users. This corresponds to the interpretation/evaluation phase of KDD.

The details of these phases in the context of movement data are covered in the remaining part of the chapter. The flow of discussion will start with preliminary concepts related to the input and the output of the GKD process, particularly applied to movement data. Afterwards, the sequence of steps that transforms the input movement data to meaningful information will be covered in the subsequent chapters. The discussion will be restricted

to semantic annotation of trajectories (which can be viewed as part of the preprocessing phase for GKD), data mining, and pattern interpretation.

2.2 The Input

As seen in Figure 2.1, the KDD process takes in data as input and applies a series of steps to it before discovering knowledge, which is the main goal of the whole process. This section describes movement data, which serves as input to a geographic KDD process. Furthermore, it also provides an overview of trajectories, which is a representation of the preprocessed movement data.

2.2.1 Movement Data

Movement data, which is also referred to as mobility data in [35], can be simply defined as a sequence of positions that a moving object goes through over time as shown in Figure 2.2.



Figure 2.2: Movement Data of an Object

As mentioned in Section 1.1, large collections of movement data are now becoming more accessible due to the latest advancements in telecommunication, wireless and location technologies. Analysis of such data can help in understanding movement behaviors of the observed entities and the surrounding movement phenomena. For instance, [14] investigates pedestrian movement in the context of mobility in carnivals and street parades where issues such as congestion or crowding are key features. Understanding crowd behaviors in this context can help in predicting and controlling congestion and other safety issues in future events.

2.2.2 Trajectories

Movement data are usually represented in the form of trajectories, which can be defined as a sequence of (x, y, t) -tuples describing its position over consecutive time instances in ascending order. (x, y) refers to the position of the moving entity at a specific time instance t .

Formally, a trajectory T over two-dimensional space is defined as a continuous mapping from

$$I \subseteq \mathbb{R} \text{ to } \mathbb{R}^2 : t \mapsto \alpha(t) = (\alpha_x(t), \alpha_y(t))$$

and

$$T = \{(\alpha_x(t), \alpha_y(t), t) \mid t \in I\}. \quad [22]$$

The given formal definition describes a trajectory as a mapping from time to space. However, time is continuous. Thus, an individual trajectory consists of an infinite sequence of time moments, wherein each moment is mapped to a position. Due to the finite amount

of memory available for storing trajectories, each trajectory should be recorded as a finite sequence. Thus, the points in a trajectory are often recorded at random time instances and for this reason, they are also referred to as sample points while the rate at which the points are recorded is referred to as the sampling rate. For example, the sequence of (x, y, t) -tuples for a specific entity may have a time gap of 5 seconds from the first recorded point to the second recorded point, a gap of 30 seconds from the second point to the third point, and so on. These points are typically measured with uncertainty and refinement is possible by considering physical constraints, such as the road network.

Interpolation Since trajectories are built only from a set of sample points in the movement data instead of building from the movement data for each time instance, there is a need to estimate the data in between the set of sample points. This estimation is known as interpolation.

The simplest and fastest type is called the linear interpolation. In linear interpolation, the sample points are connected with straight lines, which demonstrate the assumption that the speed and direction of the object's movement between every pair of sample points are constant. Despite of this, it is still a popular interpolation technique due to its simplicity and speed, both in terms of construction and handling.

Another interpolation technique that creates smoother curves in the trajectories is through the use of Bezier curves. Construction of trajectories is quite fast but handling them, such as computing the distance along the trajectory, is not as simple as with linear interpolation.

2.2.3 Complexity of Movement Data

Movement data is complex due to its multi-faceted characteristics as demonstrated in [7]. These characteristics, which affects the movement behavior of the moving entities considered, can be categorized into four main parts:

1. space component - refers to the positions traversed by the entities during the period of movement. Some examples of space-related characteristics that can influence the entities' movement behavior are the presence of obstacles like a wall, the characteristics of the surface like being made of concrete or soil, and the function of the location like being a residential or commercial area.
2. time component - refers to the time period of the entities' movement. It is worthwhile to note that time can also influence the entities' movement behavior. For example, a person moving during weekdays is most likely to move from work to home, and vice versa. This is in contrast to a person moving during weekends. In this case, he/she is more likely to move between recreational areas or to stay at home, depending on the person's preference.
3. moving entities and their activities - refers to the characteristics of the moving entities and their activities that may influence their movement behavior. Some examples of moving entity characteristics that affect movement behavior are age, gender, occupation, and health condition. An example illustrating the effect of activities on an entity's movement is by considering the speed of a person heading for work as

opposed to a person shopping in the mall. It is expected that the person going to work would move faster compared to the latter.

4. phenomena and related events - refers to the phenomena and events occurring in the moving entities' environment. Some examples are the current weather conditions, ongoing concerts, sports events, traffic accidents, and government-imposed rules.

This list confirms that while movement data is usually associated with the space and time components only (as seen in the formal definition of trajectories), the characteristics of the moving entities themselves and the surrounding events during the time of movement are important aspects of the movement data as well. For this reason, it is vital to consider this collection of relevant characteristics (i.e., semantics) in processing movement data for the purpose of obtaining meaningful knowledge, which are represented by movement patterns.

2.3 Movement Pattern

The extracted knowledge using the KDD process comes in the intermediate form of movement patterns. Dodge et al [25] describes movement patterns as regularities in terms of space and time, or interesting relations implicit in the movement data. This work has also initiated the construction of a taxonomy for movement patterns, which is shown in Figure 2.3. They have categorized movement patterns into two groups, namely, generic and behavioral patterns. Due to the large number of specific movement patterns, the succeeding examples focus on those related to flock patterns. The reader is referred to [25] for a more comprehensive and detailed discussion.

Generic patterns refer to low-level patterns that can be extracted by applying generic data mining algorithms. It is further classified into primitive and compound patterns. Primitive patterns refer to the most basic form of patterns while compound patterns consist of more than one primitive patterns. Primitive patterns are also subcategorized into spatial, temporal, and spatio-temporal patterns. An example of a spatial primitive pattern is co-location in space, which refers to a group of objects that have similar positions in space without considering time. Synchronization is an example of temporal pattern wherein similar changes of movement variables occur either at the same time or after a time delay. Meet and moving cluster are examples of spatio-temporal patterns that are closely related to flocks. A meet refers to a group of objects that stays within a stationary disk in a certain time interval while a moving cluster is described as a group of objects that moves close together in a certain time interval. A compound pattern closely related to these patterns is convergence, which is described as the movement of a set of objects headed towards the same location and is illustrated in Figure 2.4. Another related compound pattern is encounter, which is a specific form of convergence pattern wherein the objects arrive at the same time at the meeting place.

On the other hand, behavioral patterns are more complex patterns that are made up of generic patterns and are capable of describing the behavior of a specific type of moving object or a specific group of moving objects. Evasion and pursuit are some examples. These two patterns occur together since evasion describes the behavior of an animal trying to escape from a threatening and pursuing animal, while pursuit describes the behavior of the pursuing animal. The flock pattern, which is described in [25] as a group of animals

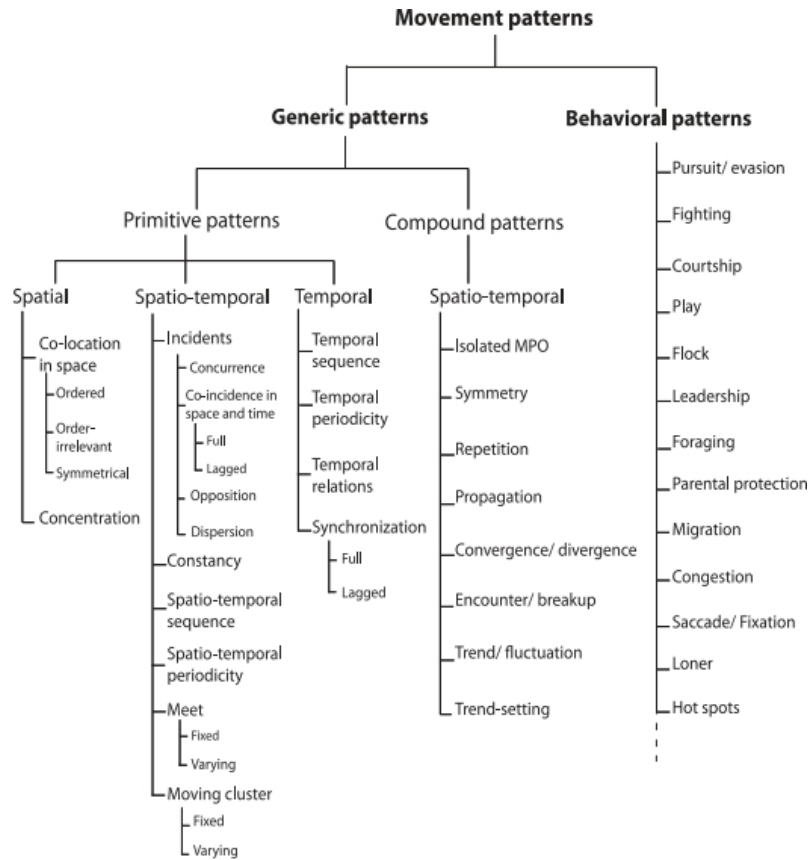


Figure 2.3: Classification of Movement Data [25]

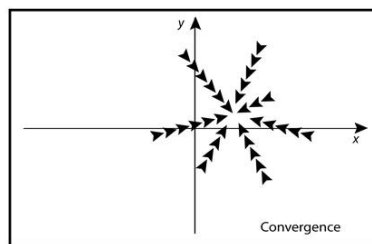


Figure 2.4: Convergence pattern. [70]

moving in the same direction while staying close together, was also classified as a behavioral pattern. It is a moving cluster set in the context of animals. Being consistent with existing data mining literature on flock patterns, we refer to flock patterns as moving cluster (i.e., flock patterns describe patterns without reference to the type of moving object and other contextual information). Since flock pattern is a central concept in the thesis, a separate subsection (refer to Subsection 2.3) is entirely devoted to existing flock definitions in data mining literatures.

Different data mining algorithms have been developed for discovering generic patterns with the aim of finding an efficient algorithm for processing large-scale movement datasets. Nevertheless, the movement patterns extracted by most data mining algorithms do not

consider the movement context, which is necessary in correctly interpreting and understanding the patterns. Clustering and flock discovery algorithms applied to movement datasets, for example, mainly use the set of sample points (x, y, t) to determine the spatial closeness of trajectories over time and often ignore context attributes, such as the age category or the job description of the moving entities, or the type of landscape wherein the movement occurred.

We chose to analyze of flock patterns since there are several existing works focusing on such patterns both in the data mining and the application-oriented literatures. Moreover, this pattern can be interpreted in different ways. Sample interpretations (i.e., behavioral patterns) that can be derived from them are pursuit, courtship, play, flock (as defined in [25]), migration, and congestion. This makes flock pattern an interesting focus of analysis. Furthermore, such analysis is useful in applications such as traffic management, and recreational management systems wherein the manager can exploit the analysis results to improve the governed area.

Since the taxonomy in [25] is a first attempt in classifying movement patterns, a joint refinement and standardization effort by researchers in related fields is still needed. In fact, the authors have setup a wiki page [62] in order to move towards this effort and this page also includes a discussion page wherein registered users can give suggestions in improving the taxonomy. We mention some of the shortcomings of the taxonomy from our point of view and from the discussion page of the wiki. One difficulty with the classification is the use of terms that have different meanings in known ongoing works. For example, flock is classified as a behavioral pattern indicating that context has been considered whereas existing literatures on flocks in computational geometry and data mining use the term flock as a generic pattern. Furthermore, the term pattern usually refers to unprocessed output in data mining. Thus, using the term interpretation instead of behavioral pattern and removing it under movement patterns may be more appropriate since the definition given for behavioral patterns includes context. Another issue that should be addressed by such a taxonomy is its completeness. Due to its data driven approach, the current taxonomy must be extended further in order to cover new movement patterns that have not yet been defined. A qualitative approach to classification of movement pattern, such as the work by Wood and Galton [92], provides this advantage over the approach used in the taxonomy of Dodge et al. However, it is still interesting to see how existing patterns can be classified and organized. It would be interesting to have a hierarchical taxonomy that shows the relations among patterns. For example, meet and moving cluster are two patterns that are closely related to each other since a meet can be seen as a special case of a moving cluster with speed equal to 0.

Flock Definitions

The following list provides a summary of flock definitions provided by existing works on flock discovery algorithms:

- *Finding REMO - Detecting Motion Patterns in Geospatial Lifelines* [51]

Flock: Concurrence with spatial constraint. Concurrence refers to a set of different entities having the same values of motion attributes (i.e., speed, acceleration, bearing/direction) for a time instance. The spatial constraint requires that entities

should not just move in the same way but should also be close to each other. Entities are considered to be close if their absolute location is within a certain radius.

Leadership: A flock with an entity that shows constance over the previous times steps. Constance refers to an entity maintaining a certain motion (i.e., constant speed, no acceleration, moving in the same direction) for a time instance.

- *Efficient Detection of Motion Patterns in Spatio-Temporal Data Sets* [38]

These are formalized definitions of those found in [51] but speed and acceleration are ignored here.

Flock: Parameters: $m > 1$ and $r > 0$. At least m entities are within a circular region of radius r and they move in the same direction.

Leadership: Parameters: $m > 1$, $r > 0$ and $s > 0$. At least m entities are within a circular region of radius r , they move in the same direction, and at least one of the entities was already heading in this direction for at least s time steps.

- *Reporting Flock Patterns* [15]

This definition emphasizes that a flock should stay together for some duration of time rather than for a single time instance, unlike in previous works. It is assumed that each trajectory has the same number of line segments.

(m,k,r)-flockA - Let $m, k \in \mathbb{N}$, and let $r > 0$ be a constant. Consider a set of trajectories, where each trajectory consists of τ line segments. A flock in a time interval $I = [t_i, t_j]$, where $j - i + 1 \geq k$, consists of at least m entities such that for every point in time within I there is a disk of radius r that contains all the m entities.

A more relaxed definition was also provided using the assumption that movements are in straight line and have constant speed between two time points with known location data. With this definition, it is no longer required to check every time point in an interval. Instead, only the time points where data has been collected need to be checked.

(m,k,r)-flockB - Consider a set of trajectories, where each trajectory consists of τ line segments. Let I be a time interval $I = [t_i, t_j]$, where $j - i + 1 \geq k$ and $i \leq j \leq \tau$. A flock in time interval I consists of at least m entities such that for every discrete time-step $t_l \in I$, there is a disk of radius r that contains all the m entities.

- *Computing Longest Duration Flocks in Trajectory Data* [37]

This definition is almost similar to that of [15], only differing in the fact that the time interval k for which the flock members stay close together could be a real number rather than just an integer.

flock(m,k,r) - Given a set of n trajectories of entities in the plane, where each trajectory consists of τ line segments, a flock in a time interval I , where the duration of I is at least k , consists of at least m entities such that for every point in time within I , there is a disk of radius r that contains all the m entities (note that $m \in \mathbb{N}$, $k \in \mathbb{R}$). They have also considered two types of flocks, namely, fixed and varying flocks. A fixed flock consists of the same m entities staying close together over the entire time interval. On the other hand, a varying flock consists of entities that may change during the interval so long as the number of entities staying close meet the minimum number requirement.

- *Reporting Leaders and Followers Among Trajectories of Moving Point Objects* [4]
The focus of this work is on identifying leaders in the flock. Hence, they have only provided a definition for *leader* and for *follows*, while referring to the same flock definition found in [15].
Leader - An entity is a leader at time $[t_x, t_y]$ iff it does not follow other entities at time $[t_x, t_y]$ and there are sufficiently many entities following it at time $[t_x, t_y]$.
Follows - e_i follows e_j iff e_j is in front of e_i (i.e., within the angle α) and $\|d_i - d_j\| < \beta$ as shown in Figure 2.5.

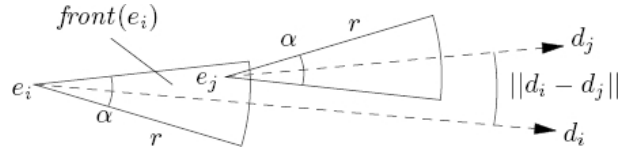


Figure 2.5: e_j is in front of e_i . [4]

- *On Discovering Moving Clusters in Spatio-Temporal Data* [47]
This work defined the concept of moving clusters, which is closely related to the notion of flocks.
Moving cluster - a group of objects moving close together for a long time interval. The objects within the group may change (i.e., addition of members, removal or replacement of some members). It is similar to varying flocks.
- *On-line Discovery of Flock Patterns in Spatio-Temporal Data* [88]
This definition is consistent with previous definitions but provides a more technical description that fits their proposed discovery algorithm.
Flock - Given are a set of trajectories T , a minimum number of trajectories $\mu > 1$ ($\mu \in \mathbb{N}$), a maximum distance $\epsilon > 0$ defined over the distance function d , and a minimum time duration $\delta > 1$ ($\delta \in \mathbb{N}$). A flock pattern $\text{Flock}(\mu, \epsilon, \delta)$ reports all maximal size collections F of trajectories where: for each f_k in F , the number of trajectories in f_k is greater or equal than μ ($|f_k| \geq \mu$) and there exist δ consecutive time instances such that for every $t_i \in [f_k^{t_1}, f_k^{t_1 + \delta}]$, there is a disk with center $c_k^{t_i}$ and radius $\epsilon/2$ covering all points in $f_k^{t_i}$.

It is important to note that these definitions and our definition of flock as well, is quite different from that of Dodge et al [25]. Recall that the definition by Dodge et al associates flocks with animals while definitions described in these works are not restricted to any type of moving entity. Moreover, Dodge et al categorize flocks under behavioral pattern, which implies that movement behaviors can be inferred from them. In this work and as with the other works mentioned here, however, flock patterns are considered as an intermediate output that require further processing in order to transform them into meaningful patterns from which movement behaviors can be inferred.

2.4 Semantic Annotation of Trajectories

In this section, we now shift the readers' focus from the input and output of the KDD process to the process of obtaining meaningful interpretations from movement data. Three

key phases for this purpose include semantic annotation of trajectories, data mining, and pattern interpretation. This section covers the discussion on the existing state of the art for the semantic annotation of trajectories. The state of the art in data mining and in pattern interpretation will be covered in the next two sections.

Before providing a survey of some tools and algorithms for semantic annotation of trajectories, it is fundamental to have a clear understanding of semantics, which is vital in delivering meaningful results to the end-user.

2.4.1 Semantics

Semantic, in the general sense, is defined in Merriam-Webster [59] as “of or relating to meaning in language.” In other words, it is related to the meaning of a considered notion.

In movement data, semantics are the information that describe the environment in which the movement occurs. Environment does not only refer to the geographical location where the movement took place. It also includes the on-going phenomena, and the other objects that the moving object interacts with. Furthermore, semantics also cover the properties of the moving object itself and the properties of the different environmental aspects.

In our study, we specifically focus on semantic data that can aid in understanding movement behavior of observed entities depending on the considered application domain. Geographical data, typical speed (i.e., reasonable range of speed values), purpose, weather condition, and age of the moving entity are some examples of semantic information. It is important to emphasize the dependence of semantics on the application context. For example, weather condition may not be as important in an employee monitoring system as opposed to a bird monitoring system.

Due to the importance of semantics in understanding movement patterns, there is a need for integrating them with the movement data and/or the extracted movement patterns. Thus, a number of semantic annotation tools were developed for this purpose.

2.4.2 Semantic Annotation of Trajectories

As mentioned in the preceding chapter, semantic annotation for the Web has been well studied as demonstrated in literatures such as [86, 78, 49, 24, 27]. On the other hand, the set of semantic annotation tools for trajectories is quite new and hence, not as rich. This section covers a subset of tools and algorithms for semantically annotating trajectories in movement data, while semantic annotation for the Web is out of the scope of the thesis.

Semantic Annotation of Trajectories with Stops and Moves

In Spaccapietra, et al [81], trajectories are viewed “as movements that correspond to semantically meaningful travels.” As a consequence, they introduced stops and moves as important semantic concepts since a travel consist of stopping in an interesting place, and moving towards and/or away from this place.

The sequence of positions over which the object continuously changes position is called a move. On the other hand, the position over which an object stays fixed for some minimum time interval is called a stop. More specifically, a stop is a part of a trajectory that has been explicitly defined by the user to represent a stop. The moving object should also stay within the stop for some non-empty time interval. Moreover, all stops are temporally disjoint (i.e., two stops should occur in different time instants).

On the other hand, a move is a part of a trajectory that is delimited by two extremities. These extremities are either two stops, or the starting point and the first stop of the trajectory, or the last stop and the ending point of the trajectory. As with stops, the time interval of a move should be non-empty.

The identification of stops and moves within a trajectory depends on the application domain. For example, stops identified for a company's tracking application would be different from the stops identified for a bird monitoring application. The following algorithms were developed for semantically enriching trajectories with stops and moves.

Stop and Moves of Trajectory (SMoT) SMoT [2] is an algorithm that converts each trajectory to a corresponding list of stops and moves. When using this algorithm, the user is expected to provide the system with the list of interesting places along with the typical time duration spent in these places. Staying in an interesting place for the specified time duration must be satisfied in order to identify the place as a stop.

In the discussion of this algorithm, the list containing the interesting place along with its associated time duration will be referred to as an application since this pair varies with the application domain. Then, each pair in the list is referred to as a candidate stop.

The algorithm basically processes one trajectory at a time. For each trajectory, the following steps are performed. Starting with the first point of the trajectory, the algorithm checks whether the point intersects the region of any candidate stop of the application. If it does, the time spent by the object within the region is computed by continuously moving to the immediately succeeding point until a point outside of the region is found. The duration of time spent by the object within the region is the difference between the time that the object exited and the time that it entered the region. If the duration is at least equal to the associated minimum duration of the region, the place is considered as a stop. Once a stop is identified, a move is then recorded between the previous stop and the recently marked stop.

Cluster-Based Stop and Moves of Trajectory (CB-SMoT) CB-SMoT [69] is an extension of SMoT. A problem with SMoT is the assumption that the user has defined all the interesting places and the typical time duration spent in each place. However, this is not always the case as the user may only know about a subset of the interesting places but not all of them. With CB-SMoT, the user is allowed to identify only a subset of all the interesting places. The algorithm itself automatically identifies places that may be relevant to the application domain.

The basic intuition that allows CB-SMoT to automatically identify interesting places is based on the following: moving objects tend to spend more time in interesting places and hence, their speed slows down in such areas. The denser part of the trajectory (i.e., the set of points that are close together in terms of space and time) corresponds to such places. An extension of DBSCAN [28] is used to cluster dense points of a trajectory into a region that is a potential stop. Once the potential stops are automatically identified, the algorithm checks the intersection of these stops with the candidate stops identified by the user. Potential stops that do not intersect with any candidate stop may still be interesting and are labeled as unknown stops. A move is generated for each part of the trajectory that is not a stop.

The main advantage of CB-SMoT over SMoT is that the former is able to identify

stops previously unknown to the user. However, in cases where speed is not relevant to the domain, SMoT may give better results. Another strong point of CB-SMoT is that it is able to generate clusters even in parts of the trajectory where some points are missing.

Other Algorithms for Stop Identification Aside from SMot and CB-SMoT, [95] also describes three approaches for identifying stops. These include velocity-based stop identification, density-based stop identification, and a time series approach called Traj-ARIMA. These algorithms do not use explicit geographical information unlike previous algorithms. Instead, they rely on intrinsic properties such as velocity and density in order to detect stops and moves. Despite of these, we classify them as semantic enrichment algorithms since they allow the extraction of semantic attributes (i.e., stops and moves), which are derived from trajectory data.

Velocity-based stop identification flags a point of the trajectory as a stop when the instantaneous speed is lower than the speed threshold. This threshold is defined with respect to the average speed of the moving object, the average speed that occurred in the nearest road crossing, and the average speed that occurred in the road segment corresponding to the point's position. A limitation of this technique is that it cannot properly handle cases wherein stops are performed by entities that move at high speed in a small area.

To handle this limitation, density-based stop identification was also introduced. This algorithm considers the maximum distance covered by an object aside from considering its instantaneous speed. Its basic idea is to consider consecutive points in a trajectory as a stop if they fall within the density area of the immediately preceding point. This area is restricted by a time duration and a corresponding maximum distance.

Finally, Traj-ARIMA is a time series approach for network-constrained trajectory modeling. It extends the Auto-Regression Integrated Moving Average (ARIMA) model with a spatial dimension and it is described in details in [94]. It is worth noting that the technique can be used for velocity fitting and prediction, aside from stop prediction.

Semantic Annotation of GPS Trajectories

Realizing the need for semantics for the analysis of movement behavior, Guc, et al [36] proposed the use GPS trajectories to facilitate manual semantic annotation without having the need for neither manual interview nor manual mobility records.

They proposed a conceptual annotation model that includes two annotation elements, which are episodes and trips. Episodes were defined by Mountain and Raper [61] as time periods in which the user's spatio-temporal behavior was relatively homogeneous while trips are sequences of episodes that are concerned with a common aim. The homogeneity of episodes in Guc, et al [36] depends on the purpose of an action and the mode of transportation though this may be extended further depending on the application domain.

Using this model, they have implemented an annotation tool developed in the Java environment. The architecture of the software includes three layers: data handling for the storage of the trajectory and annotation data, program control for the program flow, and user interface for the GUI components. The tool includes interface functionality for visualization of the GPS trajectories, display of temporal trajectory aspects through a timeline bar, trajectory animation for visualizing slow and fast movements in certain time periods and the direction of movement as well, and placemarks allowing the user to specify his/her favorite places.

Trip and Trip Purpose Extraction from GPS Trajectories

Wolf [90] and Wolf, et al [91] have studied the feasibility of replacing travel diaries, which require a manual recording and retrieval process, with automatic extraction of trips and trip purposes from GPS trajectories. Trips are automatically extracted by checking the part of trajectories wherein there is no movement detected. Once the end of trips and other relevant information are derived, the next step involves automatic extraction of trip purposes. This, however, requires a manual process of combining land use information and other geographic information in the case that the land use information is not enough. The land use data are linked with a set of purposes, which includes a primary purpose and may include a secondary or even a tertiary purpose. These combined information are used to determine the purpose of a moving entity by matching the land use with the identified purposes based on the starting and the ending positions of the trips, and the temporal component of the trips made by the entity.

Axhausen [10] proposed a similar approach that uses personal information about the moving entities' home and work addresses aside from the land use information.

Generating Semantic Annotations for Frequent Patterns with Context Analysis

The work described in [58] proposes an approach for automatically generating semantic annotations for frequent patterns. This is realized by building a context model, extracting representative transactions, and finding semantically similar patterns for each frequent pattern. The context model is built by selecting a set of informative context indicators, which is made up of context units that have the strongest weights with respect to the currently considered frequent pattern. Each context unit carries semantic information and should co-occur with some pattern. Furthermore, a context unit can be an item in a transaction, a pattern, or a whole transaction. Redundancy within the set is eliminated through a microclustering technique wherein redundant units are clustered together. The representative transactions are extracted by modelling each transaction as a vector that is similar to the vector representation of the frequent pattern's context model. Then, the cosine similarity is used to compute the similarity between each transactions and the context model. The top-ranking transactions based on this measure are chosen as the frequent pattern's representative transactions. Finally, the set of similar patterns are selected by computing the similarity between the context model of the frequent pattern with that of the candidate patterns.

2.5 Data Mining

Once movement data has been enriched with semantics, the next step in the KDD process is data mining in which the main purpose is to discover movement patterns hidden in the data. The succeeding discussion on data mining is mainly split into two parts, one part focusing on the classical data mining algorithms and another part focusing on the data mining algorithms specifically used for movement data.

2.5.1 Classical Data Mining Algorithms

This section provides an overview of the three main data mining tasks, namely, classification, clustering and association mining as discussed in [83].

Classification Algorithms

Classification refers to the “task of learning a target function t that maps each attribute set x to one of the predefined class labels y .” [83] It is performed for either one of the following purposes: descriptive modeling (to support the explanation of why objects belong to a specific class), or predictive modeling (to predict the class membership of objects).

A specific type of classification algorithm called decision tree induction algorithm extracts a series of conditions based on the attributes in the dataset for classification. It basically splits the data by using the attribute that minimizes data impurity (i.e., positive examples of the target class are well separated from the negative examples). This is recursively applied to the smaller subsets of data resulting from previous splits until all instances in a subset belong to the same class.

The classification algorithm we used for the pattern interpretation framework is J48 [76, 45], which is a type of decision tree induction algorithm. It is an implementation of the well-known C4.5 [75] decision algorithm for generating a pruned or unpruned decision tree. It uses normalized information gain in order to select the attribute for splitting the data. It can handle both continuous and discrete attributes, data with missing values and attributes with differing costs. It also provides an option for pruning generated trees.

This type of algorithm has the following advantages: they are non-parametric (i.e., makes no assumption about the probability distribution of attributes in the dataset); tree construction is computationally inexpensive and classification can be quickly performed once the tree model is built; the generated trees are easy to interpret and their accuracy is comparable to other techniques; it is robust to the presence of noise; redundant attributes do not adversely affect the accuracy of the generated decision tree though feature selection techniques can help if there are too many irrelevant attributes; the choice of impurity measure has minor effect on its performance. Its weaknesses include the following: decision trees do not generalise well to certain types of Boolean problems; leaf nodes may correspond to decisions that are not statistically significant but this may be solved by disallowing further splitting when the number of records is below a threshold; a subtree may be replicated several times in a decision tree; use of oblique decision tree or constructive induction may improve the expressiveness of the decision tree, which is restricted to rectilinear decision boundaries when splitting is only based on a single attribute.

There are many other types of classification algorithms. An example of which is the rule-based classifier, which is as expressive as a decision tree. The Bayesian classifier is another group of classifiers that are statistically-based and founded on Bayes’ theorem. More sophisticated classifiers are artificial neural networks (ANNs), and support vector machines (SVMs). ANNs are inspired by how biological neural systems work while the foundation of SVMs is based on statistical learning theory. A detailed discussion of these algorithms is not provided in this work since these are not within the scope of the thesis.

Clustering Algorithms

Clustering refers to the task of dividing the data into groups (i.e., clusters) wherein objects belonging to a group are more similar to each other compared to those belonging to other groups. It is performed for the purpose of understanding the observations in the dataset by determining the group of objects that share common characteristics when the class labels are not known beforehand. It is also used for finding the most representative cluster prototypes which are helpful for specific applications, such as summarization, compression and efficiently finding nearest neighbors.

Types of Clusterings There are different types of clusterings and they can be generally categorized as follows: partitional vs. hierarchical, exclusive vs. overlapping vs. fuzzy, and complete vs. partial. Partitional clustering divides objects into non-overlapping clusters while hierarchical clustering divides objects at different levels, finding both clusters and subclusters. Exclusive clustering assigns an object to a single cluster while overlapping clustering allow an object to belong to more than one cluster. On the other hand, fuzzy clustering allows each object to belong to every cluster with a membership weight ranging from 0 to 1. Complete clustering assigns all objects to some cluster while partial clustering may not assign some objects to any cluster at all.

Types of Clusters Aside from having different types of clusterings, there are also different types of clusters. General category for such types includes the following: well-separated, prototype-based, graph-based, density-based, and shared property clusters. Well-separated clusters refer to groups of objects such that the distance between objects belonging to similar cluster is smaller than those belonging to different clusters. Prototype-based clusters are sets of objects in which objects belonging to the same cluster are closer (i.e., more similar in terms of a specific set of criteria) to the prototype defining the cluster compared to the prototype of any other clusters. If the data is represented by a graph wherein nodes refer to objects while edges refer to connections among the objects, a graph-based cluster is a connected component, which is a group of objects that are connected to each other but not to other objects that are outside of the group. An example of a graph-based cluster is a contiguity-based cluster wherein two objects are connected if they are within a specified distance of each other. This implies that in this type of clusters, each object is closer to some other object within the cluster compared to any point belonging to another cluster. A density-based cluster is a dense region of objects that is surrounded by a low-density region. A shared property cluster is a group of objects that share some property. It encompasses the previous types of clusters and includes other new types of clusters as well.

Representative Clustering Algorithms There are different varieties of clustering algorithms but this section only describes three representative algorithms, which employ different types of clusterings and produces different types of clusters. These are k-means, agglomerative hierarchical clustering, and DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

K-means [55] is a prototype-based and partitional clustering algorithm. It finds k clusters, where k is a user-specified parameter. Each cluster is represented by its centroid, which is usually computed as the mean of the group of points. It is worthwhile to note

that this algorithm is very similar to k-medoid algorithms, wherein a medoid referring to an actual central point is used to represent each cluster. It has the advantage of simplicity and can be used for a wide range of data types. Moreover, it is also quite efficient, although several runs are often performed. Variations of the basic algorithm are less susceptible to initialization problems. However, it is not suitable for non-globular clusters, or clusters with different sizes and densities. It is restricted to data for which the notion of a centroid exists. Outliers can also pose a problem in the clustering results.

Agglomerative hierarchical clustering, which is the clustering technique we employed in the pattern analysis step of the proposed framework, produces the clustering result by initially considering each point as a singleton cluster before recursively merging the pair of clusters that are most similar to each other until an all-encompassing cluster remains. It applies a technique opposite to that of a divisive hierarchical clustering, which uses a top-down approach instead.

The result obtained from hierarchical clustering distinguishes it from other clustering algorithms. While other algorithms usually produce only a single level of clustering result, hierarchical clustering produces a result with different levels of clustering. In particular, its result is often displayed as a tree-like diagram called a dendrogram, which shows the cluster-subcluster relationships among the points and the order in which the clusters were merged (for agglomerative) or split (for divisive). The result can also be represented by a nested cluster diagram, wherein clusters containing the points are represented by circular regions, which may overlap. Having this type of nested result is advantageous because it provides the additional information of how individual clusters are related to other clusters. This information is not provided by a single level of clustering result.

Typically, agglomerative hierarchical clustering is utilized in applications requiring a hierarchy, such as constructing a taxonomy for different species of birds. Some studies have suggested that this type of clustering produces better-quality clusters. However, it is expensive in terms of computational and storage requirements. In addition to this, local merges performed at each step of the algorithm are final, which can be a problem for noisy and high-dimensional data. These can be resolved by performing partial clustering using another algorithm before finalising the results using this algorithm.

DBSCAN [28] is a partitional clustering algorithm wherein the number of clusters is automatically determined. With this algorithm, points in low density regions are considered as noise and are therefore, omitted. Hence, it is not a complete clustering algorithm. The density of a point is estimated based on the number of points that are within a specific distance Eps from it.

DBSCAN's strength includes its relative resistance to noise and its ability to handle clusters of arbitrary shapes and sizes. It has trouble, however, in finding clusters that have varying densities. It can also be expensive when the nearest neighbor computation requires all pairwise proximities, which is often the case.

Association Rule Mining Algorithms

To complete the discussion on classical data mining algorithms, a brief overview of association rule mining is provided. Even though we did not use it in the proposed pattern interpretation framework, future works may exploit such type of algorithms.

Like classification and clustering, association analysis is another data mining task useful for discovering interesting relationships hidden in large quantities of data. In this

case, the discovered relationships are represented in the form of association rules. An association rule is an implication expression having the form $X \rightarrow Y$, where X and Y are disjoint itemsets (i.e., set of items corresponding to a subset of attributes found in the dataset). Each association rule should satisfy the minimum support and minimum confidence requirements. Support is the number of occurrences of the rule in the input dataset and it is often used to eliminate uninteresting rules. Meanwhile, confidence is a measure of the reliability of the inferred rule. It is important to understand that association rules do not necessarily imply causality but simply suggest a strong co-occurrence between the antecedent and the consequent of each rule. Implication of causality requires knowledge about causal and effect attributes.

2.5.2 Mobility Data Mining

From general data mining algorithms in the KDD process, this section shifts the discussion to data mining algorithms that focus on pattern extraction from movement data.

The main challenge today is turning movement data into useful information. Although several data mining techniques already exist, most of these techniques were designed without having the complexity of movement data in mind. To address this problem, a novel area of research known as mobility data mining is emerging. It aims to analyze movement data through efficient extraction of appropriate patterns and models; “it also aims at creating a novel knowledge discovery process explicitly tailored to the analysis of mobility with reference to geography, at appropriate scales and granularity.” [35] Since mobility data mining deals with movement data that is set in a specific geographic location and geographic information is an important semantic concept in this field, it can be seen as a step corresponding to the knowledge extraction step of the GKD process.

The discussion of mobility data mining algorithms is split into two parts, namely, spatio-temporal clustering and semantic trajectory data mining. We have decided to put spatio-temporal clustering under mobility data mining since it deals with data having spatial and temporal components, which is the case for movement data. On the other hand, semantic trajectory data mining specifically deals with movement data and their semantics and for this reason, we have placed them under mobility data mining as well.

Spatio-Temporal Clustering

Spatio-temporal clustering, which is a specific clustering type that groups together objects based on their spatial and temporal similarity, is becoming popular especially in the field of geographic information science. This is due to the fact that movement data recorded based on location technologies are becoming more widely available, and these recorded data contain both spatial (i.e., location of the moving objects at specific time instances) and temporal (i.e., period of movement) components. Since this thesis focuses on flocks, which is an instance of spatio-temporal pattern in geographic space, we limit the discussion to patterns related to a geographical context and ignore patterns in other domains, such as in biological or chemical. The work in [50] provides a survey of spatio-temporal data types and clustering methods for trajectory data, and we use this as a main reference for the discussions in this section.

Spatio-temporal Data Types Spatio-temporal data types for point-wise objects can be classified into the following categories as described in [50]: ST events, geo-referenced variables, geo-referenced time series, moving objects, and trajectories. Each ST event is associated with its location and its corresponding timestamp, and both the spatial and the temporal components are static. A geo-referenced variable, on the other hand, refers to “the time-changing value of some observed property.” In this case, the object remains in a specific location and a snapshot of some interesting phenomena is updated over time. Closely related to geo-referenced variables are the geo-referenced time series, wherein the whole history of the observed property’s evolution over time at a specific location is stored. Meanwhile, we are dealing with moving objects when the spatial component changes as well aside from time and only the most recent location of the object is kept. When the entire history of the object’s location over time is store, this spatio-temporal sequence forms a trajectory.

Categories of Clustering [50] categorized clustering methods for trajectory data as follows: descriptive and generative model-based clustering, distance-based clustering methods, density-based methods and the DBSCAN family, visual-aided approaches, micro-clustering methods, flocks and convoys, important places, and borderline cases.

Descriptive and generative model-based clustering methods aim to find a model describing the whole dataset. Some examples of such methods are found in [31], [20], and [1].

Distance-based clustering methods basically employ a distance or a similarity function before utilizing general clustering techniques that groups together objects with small computed distances. Some examples of distance functions on the trajectory domain are found in [64] and [72].

Density-based methods and the DBSCAN family separates high density regions from low density regions in order to identify clusters and noise points. Some examples of which are DBSCAN, which was already described earlier and OPTICS (Ordering Points To Identify the Clustering Structure).

Meanwhile, visual-aided approaches attempt to overcome issues, such as finding patterns that are trivial or incorrect within the considered context, coming with automatic algorithms.

Micro-clustering methods aim to group together trajectory segments representing objects that are spatially close over a maximal time interval. [43] use piecewise segments to represent trajectories and determines the maximal time interval for which all the trajectories are pair-wise close to each other. [53] represents each cluster as a bounding rectangle containing close trajectory segments co-occurring at similar time intervals. Meanwhile, [52] uses a density-based clustering method to group such segments.

Moving clusters, flocks and convoys address the need for discovering group of objects that move together for a minimum period of time. [47] introduced the notion of moving clusters, which refer to a sequence of clusters consisting of spatially close objects that may leave or enter the cluster during a certain time period in which the number of objects in the cluster satisfies the given threshold. Sample works, such as those found in [37] and [88] deals with a more specific type of moving clusters wherein the same set of objects stays together within a circular region of a specified radius and this are known as flock patterns. Since this is a central concept in the thesis, two succeeding subsections are devoted to

the discussion of existing flock discovery algorithms and a comparison framework for such algorithms. Another type of moving cluster called a convoy pattern is described in [46]. In this case, the same set of objects stay together in a region of arbitrary shape and extent.

Clustering approaches were also used to identify interesting places in input trajectories. Some of these works are found in [48], [2], [97], and [69]. The approach proposed in [48] computes the distance between the current location and the previous location. If the distance is smaller than the threshold, the current location is clustered with the previous location. Otherwise, the current location is added to a newly created candidate cluster. This candidate cluster is considered as a cluster if the time difference between the first point and the last point of the cluster is greater than the time threshold. Similar ideas were used in [2] and [97], while [69] uses speed characteristics to identify interesting places.

Two borderline cases that deal with trajectory patterns are described in [50]. The first one, which is found in [34], presents a grid-based clustering algorithm for finding frequent movement patterns that represent cumulative behavior of moving objects in the dataset. This type of pattern is called T-pattern and it consists of a sequence of regions with temporal transitions linking each pair of consecutive regions. Each region is specifically referred to as a region of interest, since they describe areas that are relevant and interesting to the application domain. These regions can be statically defined based on domain expert knowledge or they can be automatically computed by identifying dense regions wherein several moving entities have occurred. Figure 2.6 presents an example of a T-pattern with several regions of interest and representing 2 actual T-patterns that only differs in travel time. The figure represents the aggregate downward movement along the 4 regions represented by rectangular areas. The 2 lists of numbers on the right-hand side represents the 2 popular travel time between the regions. The labels 0, 1, and 2 refer to the typical travel time for each pair of regions of interest. For example, 0 56.4, 60.61 indicates that the travel time from the first region (i.e., topmost) to the second region is ranges from 56.4 to 60.61. The other work found in [44] proposes an improvement to the previous grid-based approach by partitioning trajectories into disjoint segments that represent meaningful spatio-temporal changes in the object's movement, and applying a clustering algorithm to group similar segments.

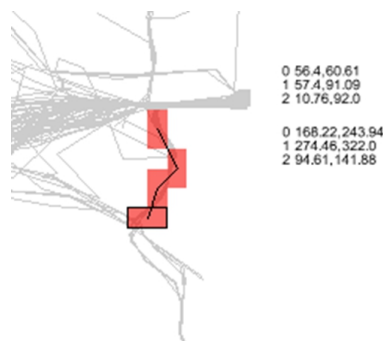


Figure 2.6: An example of a mined T-pattern.

Survey of Existing Flock Discovery Algorithms

This section provides a survey of existing flock discovery algorithms, which fall under spatio-temporal clustering algorithm. This discussion is important since flock discovery is also a central concept in the thesis, particularly for the first step of the proposed pattern interpretation framework.

A pioneering work in flock detection is the Relative Motion (REMO) framework proposed by Laube et al [51]. In their work, the motion pattern of each entity is described in terms of speed, change in speed and azimuth vector (i.e., bearing or direction). In this algorithm, a flock is defined as a set of entities having similar motion attributes and stays spatially close at some time instance. To find flock patterns, entities having similar motion pattern at some time instance are grouped together. These groups are further split into clusters, each one containing objects that are spatially close based on their point coordinates. The work, however, can be extended further by providing a mechanism for finding flocks that last for a time interval rather than a single time instance.

The work by Gudmundsson et al [38] provides an efficient algorithm for finding flock and other related patterns. To measure the spatial closeness of objects, their algorithm utilizes a compressed quadtree, which is a tree-like data structure that recursively splits the data into 4 parts and causes spatially close points to belong to the same cell. Each quadtree stores the point coordinates at a specific time instance. An arrangement is built based on each quadtree and a disk is rotated around the arrangement. Any disk containing the minimum number of entities is considered as a flock. But as with the previous algorithm, further extension is necessary to discover flocks lasting for more than one time instance.

Benkert et al [15] broadens the flock definition in the previously described works by emphasizing that the entities should stay together for a period of time, say k , rather than for a single time instance. To discover flocks, the given 2-D points are first transformed to 2k-D points (i.e., from $\langle (x_i, y_i), (x_{i+1}, y_{i+1}), \dots, (x_j, y_j) \rangle$ to $(x_i, y_i, x_{i+1}, y_{i+1}, \dots, x_j, y_j)$). To determine the 2k-D points that are close to an (x, y) pair, a pipe consisting of 2k-D points is constructed for each time instance such that each pipe contains the 2k-D points that are close to the (x, y) pair at a specific time instance. A flock is found if the intersection of k pipes that corresponds to k adjacent time instances contains the minimum number entities.

Moreover, Benkert et al [15] have proposed 3 approaches to minimise the number of reported flocks. The first approach marks members of already discovered flock so that they can no longer be considered as members of other flocks. This approach, however, is too restrictive and causes loss of many interesting patterns. Our approach for handling redundancy is a variation of their second approach, which disregards entire trajectories as base if they were already included in some flock. This is still quite restrictive therefore we have modified it such a way that only trajectory segments belonging to some flock are disregarded as base, and not the whole trajectory. Finally, their last approach extends discovered flocks by joining them. Nevertheless, this is a more time-consuming approach compared to the others.

In scenarios wherein the user does not want a fixed time interval k , flocks can be defined as having an m number of entities staying close together for the longest possible time period. Gudmundsson and van Kreveld [37] propose finding longest duration flocks by building a cylindrical volume for each entity. Every volume is built by considering

every point in the entity's trajectory and computing the entities that are spatially close to it for each time instance. Then, the time interval for which other entities are inside the volume is computed and sorted by their starting point in non-decreasing order. All intervals starting at the earliest time t_1 are collected and sorted based on their ending point. The $(m - 1)$ th time interval from the last endpoint determines the interval of the longest duration flock that starts at time t_1 . The next earliest starting point t_2 is then considered. Time intervals with endpoints between t_1 and t_2 are removed before repeating the same process as with t_1 . Same procedure is repeated for the rest of the starting points.

Kalnis et al [47] presents an algorithm for finding moving clusters, which are closely related to flock patterns. A moving cluster is a group of entities that move close together for a long time period. The entities composing the moving cluster may change over time. To find such clusters, DBSCAN [28] is used to cluster the points for each time slice. Similar clusters in adjacent time slices are then combined by checking the intersection of objects in the clusters.

In Andersson et al [4], the leader of the flock pattern is determined by manipulating a set of arrays that describe whether each entity follows any other entities for the considered time interval and other related information. They define a leader as an entity moving in front of a sufficient number of entities for a long time period.

A most recent work by Vieira et al [88] describes a set of algorithms for discovering flock patterns in an on-line setting (i.e., when data is not archived but streamed part by part). The basic algorithm computes disks that determine all the possible groupings of objects for each time instance, and then merge disks in adjacent time slices if they have the minimum number of objects in common. Additional filtering algorithms are proposed in order to shorten execution time.

We found the work of Gudmundsson and van Kreveld [37] and Kalnis et al [47] interesting for the purpose of mining moving flock patterns. However, the algorithm proposed by Gudmundsson and van Kreveld [37] has not been implemented so far. As for the algorithm proposed by Kalnis et al [47], it was implemented but the algorithm performs an approximation in checking the common members of moving clusters for adjacent time slices, and this approximation introduces a disadvantage since members of a moving cluster may entirely change over time. In order to avoid this issue, the corresponding threshold could be set to 100% but this leads to loss of several interesting patterns since 100% implies that members of moving clusters in adjacent time slices should have exactly the same members.

A Comparison Framework for Flock Discovery Algorithms

Wood and Galton [92] have proposed a taxonomy for collective phenomena in which a set of criteria are described as one of the following:

- Membership - refers to the identity and cardinality of the members belonging to a collective.
- Location - refers to the location of the members, the location of the collective if applicable, and the relation between these two.
- Coherence - refers to the source of the coherence, which can be defined as the attributes or behavior of the collective as a whole rather than the attributes of each

member in the collective. For the purpose of discovering flock patterns, we have introduced two sub-categories of this criterion:

- Spatio-temporal Coherence - refers to the spatial closeness of the flock members over some time interval
- Moving Coherence - the movement of the members of a flock should not be stationary. This sub-category indicates that the flock members are indeed moving together, and not simply stopping together.
- Roles - refers to the fact that members of a collective may or may not be differentiated by role, which could be some function or position.
- Depth - refers to the possibility of some members to be collectives themselves.

We have used this set of criteria to compare the approaches proposed in the literature for discovering flock patterns. Table 2.1 summarizes the attributes considered by different flock discovery algorithms based on these criteria for collective phenomena. Since none of the approaches utilize the depth criterion, we have disregarded it from the table.

<i>Approaches</i>	<i>Membership</i>	<i>Location</i>	<i>Spatio-Temporal Coherence</i>	<i>Moving Coherence</i>	<i>Roles</i>
Laube et al [51]	Based on speed, change in speed, azimuth vector, and spatial closeness	Considers member locations	Only for one time instance	Extracts both moving and stationary flocks but do not distinguish between them	Distinguishes leaders from followers
Gudmundsson et al [38]	Based on azimuth vector and spatial closeness	Ibid.	Ibid.	Ibid.	Ibid.
Benkert et al [15]	Based on spatial closeness	Ibid.	Spatial coherence over some time interval	Ibid.	N/A
Gudmundsson & van Kreveld [37]	Ibid.	Ibid.	Ibid.	Ibid.	N/A
Kalnis et al [47]	Ibid.	Ibid.	Ibid.	Ibid.	N/A
Andersson et al [4]	Based on spatial closeness and bearing	Ibid.	Ibid.	Ibid.	Distinguishes leaders from followers
Vieira et al [88]	Based on spatial closeness	Ibid.	Ibid.	Ibid.	N/A
Our approach [89]	Ibid.	Ibid.	Ibid.	Prunes out stationary flocks; may find flock patterns whose members stop for some time	N/A

Table 2.1: Comparison of flock discovery approaches using the taxonomy provided by Wood and Galton [92].

All the mentioned approaches use spatial closeness to determine the members of a flock. Laube et al [51], Gudmundsson et al [38] and Andersson et al [4] check other motion attributes to determine flock membership. Location is a main factor for finding flock patterns in all approaches. Coherence over a time duration is considered in all approaches, except for [51] and [38]. The remaining approaches approximate coherence over time by considering discrete time steps, except for Gudmundsson & van Kreveld [37], which describes coherence over a continuous time series. [51], [38] and [4] distinguish leaders from followers. None of these approaches took depth into consideration.

Although the Table 2.1 shows that most of the proposed approaches are quite similar in terms of their conceptualization, they considerably differ in terms of their mining steps.

Among approaches that consider spatial coherence over some time interval, our approach has a merging step also found in Kalnis et al [47] and Vieira et al [88]. The mining step proposed by Kalnis et al [47] only approximates the similarity between flock memberships of adjacent time instances since they are dealing with moving clusters. The mining steps proposed by Vieira et al [88], on the other hand, use a greedy approach in the merging step of their "Basic Flock Evaluation" algorithm. They compare flock members of adjacent time slices in the following manner: initially compare the first 2 adjacent time slices, then the intersection of members in these time slices are compared with the members of the 3rd adjacent time slice; the intersection of the members of the first 3 adjacent time slices are then compared with the members of the 4th time slice and so on. This can lead to splitting longer duration flock patterns into shorter duration patterns. Instead, we use a more exhaustive approach for merging in order to avoid the loss of longest duration flocks.

Semantic Trajectory Data Mining

Semantics has been typically exploited in the field of text mining, but more recently, researchers in other mining sub-areas have acquired an interest in the development of methods that exploit semantics in the discovery process. Likewise, semantic data mining is usually associated with the Web application domain although it is a general field applicable in different domains. It is a relatively new research field, especially in non-Web application domains such as geography, that is generally concerned with the extraction of meaningful information from semantically-enriched data. A growing group of researchers is becoming active in this general field as demonstrated by a semantic data mining workshop called *Semantic Aspects in Data Mining*, which is not restricted to Web applications. This workshop has recently held its third edition and its "key idea is to develop a more general understanding about how to exploit data semantics and background knowledge, and to create standardized procedures for designing more intelligent data mining." as stated in [79].

Semantic trajectory data mining is a specialized field of semantic data mining that deals with movement data represented using trajectories. There is only a small number of research works in this relatively new field and we describe one such work.

Alvares et al [3] presented a framework for semantic trajectory knowledge discovery, which is shown in Figure 2.7. In order to extract meaningful patterns, both raw data containing observation points and raw geographic data are preprocessed and the processed data are integrated in order to obtain semantically enriched trajectory data. Specifically, the trajectory data are enriched with stops and moves. This data, in turn, goes through the data mining phase of the knowledge discovery process in order to extract interesting patterns, which are more meaningful compared to patterns extracted from trajectory data that are not enriched with geographic information.

In fact, this work describes a specialized field of the GKD process called semantic trajectory knowledge discovery. At the same time, the data mining part of the framework presents the existing state of the art in semantic trajectory data mining wherein most works focus on geographic information as semantics. It is for this reason that we categorized it under semantic trajectory data mining.

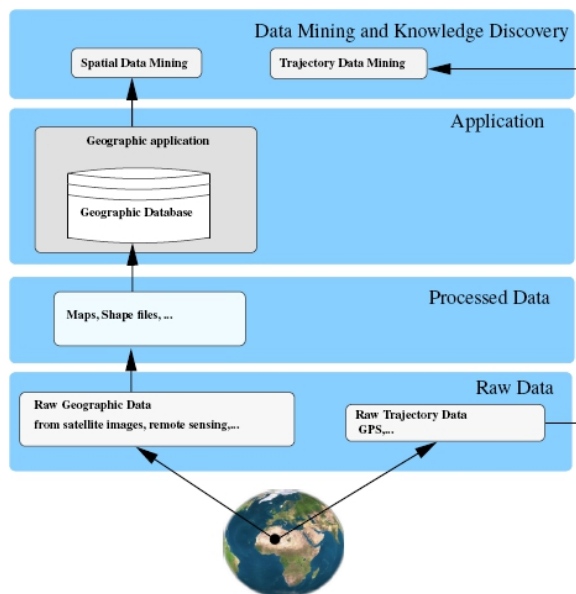


Figure 2.7: Proposed framework for semantic trajectory knowledge discovery. [3]

2.6 Pattern Interpretation

This section provides a discussion of the state of the art in the area of pattern interpretation. It is split into two main parts: the use of visual analytics tools and ontology-based systems.

2.6.1 Visual Analytics Tools

Visual analytics tools, such as those found in [5] and [18], have been developed to address this problem by allowing patterns to be superimposed against a geographical map, which represents the semantics of the locations. For example, the type of the roads (ex: highway, railroad track, service road) is color-coded in the map. Such tools aid the user in identifying the interesting places wherein the patterns have occurred, and in filtering trajectories/patterns based on some geometrical properties (ex: patterns that are spatially or temporally close to a selected pattern, patterns occurring in certain locations, patterns occurring at specific time periods).

[18] describes an object-oriented and GIS-based system that facilitates exploration and spatial analysis of household level activity-travel behavior. It has been tested using data from Portland Metro's 1994/1995 Household Activity and Travel behavior survey [82] (includes comprehensive description of activity/travel behavior of 4,451 households), Metro's Regional Land Information System (provides information about regional traffic analysis zones and political boundaries), and the 1990 U.S. Census (for assigning households to either urban or rural space). These data were integrated using an OOAD approach, resulting in the construction of a database model, consisting of spatial and non-spatial classes, using the Unified Modeling Language (UML) [65]. The model was implemented using ESRI's ArcGIS suite [8] and its Geodatabase [32] data model.

The system provides a set of exploratory tools that allows the analyst to explore the

activities of household members, represented as spatial point patterns, residing in a selected set of areas during the chosen time period. These tools allow the exploration of the central tendency (by generating a bivariate, unweighted, mean center), the dispersion (using a measure of standard deviational distance and by constructing a standard deviational ellipse), a convex hull (using an ArcObjects hull method to a geometry object containing locations of the household's activities) of the household members' activity patterns, and the visualization of the activities as an assembly of space-time paths. The central tendency and the dispersion describe the most important location for the households considered and the manner in which they leave this central location. Meanwhile, the convex hull allows the analyst to determine if the households' locations fall in urban or rural areas. Finally, the space-time path is useful in examining interactions among the individual household members.

[6] provides a guideline for the development of visual analysis toolkits that support visual exploration and analysis of large scale movement data. It emphasizes the importance of combining visual displays, database technologies, and computational methods for data processing and analysis since visualization enhances the analyst's perceptual and cognitive abilities while database operations and computational methods handle the mechanical processing aspect of large trajectory data. The authors provided a list of pattern types that can be extracted from movement data, along with the set of methods and techniques that can support the analysis of each type. Analysis of an individual entity over time, a group of entities at a specific time instance, and a group of entities over time were considered for exploration.

Adapting the multidisciplinary approach described in [6], a movement analysis framework along with a set of implemented tools is introduced in [5]. The framework combines interactive visual displays that support human cognition and reasoning, with database operations and computational methods to handle massive movement data. It consists of the following steps: data processing, extraction of significant places, extraction of trips, and examination of trips. Once the raw movement data has been preprocessed, significant places are extracted by automatically detecting the places, called stops, where the moving entity has stopped for a minimum time interval. Repeated stops and occasional stops are differentiated through the application spatial clustering on them. With the use of a map and through the analysis of temporal distribution of the stops, significant places are identified. Based on these extracted places, trips (i.e., movement from one significant place to another) are identified by dividing the movement data by temporal gap, by temporal cycles, by spatial gap, and/or by specified places. Finally, the extracted trips can be examined at different granularity, namely, individual trips, cluster of similar trips, and multiple trips.

Such tools have indeed helped the users move a step closer to understanding patterns but these can be further enhanced by considering other types of semantic data, such as specific characteristics of the moving entities, aside from considering geographical data.

2.6.2 Ontology-based Systems

Only a few research works have exerted effort in explaining the occurrence and the nature of the extracted patterns based on other available semantic information, aside from geographical data. An example of which is found in [11], wherein frequent patterns based on the discovered stops are postprocessed in order to classify patterns according to a pre-

defined domain ontology. The ontology represents the concepts, rules and assumptions present in the considered application domain.

[11] provides a model for conceptual representation and deductive reasoning of trajectory patterns obtained from raw movement data as shown in Figure 2.8. This is achieved by semantically enriching raw trajectories with semantic information, such as stops and moves. A selected data mining algorithm is then applied on the enriched trajectories. Finally, an ontology, which includes the specification of semantic trajectory concepts, domain knowledge concepts (including geographical knowledge, etc.), and behavioral patterns, is exploited in order to classify the mined patterns based on the movement behaviors, the related concepts and the axioms defined in the ontology. The specification of concepts and axioms related to semantic trajectories and movement behaviors allow the definition of complex movement behaviors, which are used as target attributes for the classification process. This model was initially introduced in [12], and it was refined further in [11].

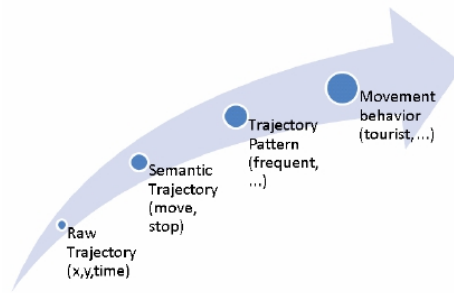


Figure 2.8: The trajectory semantic enrichment process. [11]

[11] also demonstrated the feasibility of the model by describing a specific implementation called Athena whose system architecture is illustrated in Figure 2.9. The raw trajectories are stored in the Oracle-based moving object database called Hermes [73]. A set of stops were computed from the trajectories considering a simplified geographical domain (contains museums, theatres, universities, hotels, B&B, and monuments) and storing the stops in a table. A frequent pattern algorithm is then executed with the table of stops as input and the results stored in the frequent patterns table. The defined ontology is imported from Protégé [74] to Oracle 11g [67], and the tables are translated to RDF triples before being stored as instances in the ontology. Finally, the reasoned is executed to infer new triples, which may give interesting interpretations of the mined frequent patterns.

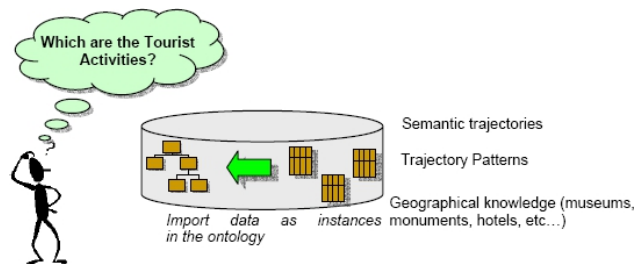


Figure 2.9: Overview of the Athena system architecture. [11]

This system has been integrated to Daedalus [68], which is a data mining query lan-

guage for spatio-temporal data, providing a system that provides support for the entire knowledge discovery process. The integrated system is called DAMSEL [84], which allows progressive querying of semantically-enriched data and movement data. Figure 2.10 shows a sample analysis result that can be obtained using the system. It illustrates the trajectories of a group of commuters that have a common starting point in space and have varying destinations, as indicated by the small squares. These destinations may group together using grids. It is worth noting that not all trajectories in the input dataset exhibit commuter behavior but through proper encoding this concept and related concepts as well in the ontology, it was possible to classify a subset of the trajectories as belonging to commuters.

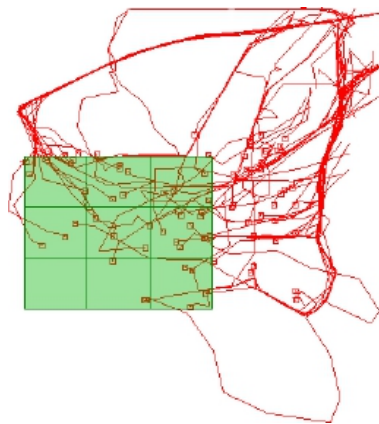


Figure 2.10: Analysis of Commuter destinations. [84]

2.7 Summary and Conclusions

This chapter provided an overview of the KDD process in general, and the different aspects of the process when applied to movement data, which falls under the specialized KDD process called GKD. The input to the process is the movement data while the output is the information needed by the end-user. An intermediate form of this output is the movement pattern, which is a representation of regularities or interesting relations in the movement data. In order to transform movement data to information, such as movement behavior, it must undergo the sequence of KDD/GKD steps. These steps includes three phases, which are preprocessing, data mining, and interpretation/evaluation. The preprocessing phase typically involves cleaning the movement data, selecting the relevant observations and attributes in the data, and reconstructing the data into trajectories. This phase also encompasses semantic annotation wherein concepts describing the movement context are integrated into the movement data. Then, data mining deals with the preprocessed movement data and extracts movement patterns from it. Finally, these patterns are analyzed and interpreted during the interpretation/evaluation phase in order to provide the information that is meaningful and worthwhile to the end-user.

The discussion on existing state of the arts in data mining demonstrate that research in classical data mining is quite mature while those in mobility data mining, which focus on movement data, is still new. Though some classical data mining algorithms may also

be applied in mobility data mining, further developments are still necessary in order to handle the complexity of movement data compared to other forms of data and to deal with analysis tasks relevant in the movement context compared to the analysis of general forms of data. Spatio-temporal clustering, which can be categorized under mobility data mining, is only an initial step since it only considers two important components of movement data, which are the space and time components. More sophisticated algorithms that consider moving entities and their activities, and phenomena and related events should be designed and developed. The other category of mobility data mining, referred to as semantic trajectory data mining, specializes in movement data represented as trajectories. Since it is a new field, we were only able to present one work [3] that fits this category. In fact, this work towards semantic trajectory knowledge discovery describes a specialized GKD process that takes semantics into account in processing trajectories. At the same time, its data mining phase is a form of mobility data mining that mainly considers geographic data as semantics, which is the general case in existing research on mobility data mining.

It is also described in the chapter that most research efforts have been concentrated in the data mining phase of the KDD process while giving less attention to an equally important phase, which is the interpretation/evaluation phase. Joint research efforts in mobility data mining and pattern interpretation is starting to address this issue. However, these research efforts mainly focus on geographic data for semantics and pattern interpretation techniques is currently limited to visualization- and ontology-based approaches.

Furthermore, the use of semantics in data mining has been often associated with the Semantic Web and few literatures on semantic trajectory data mining exist. More research effort should be given to this specialized field since results obtained for the application of techniques in this area are useful in applications such as recreational management, traffic management or animal monitoring applications.

In order to address a subset of these discussed issues, we propose a framework that considers other semantic attributes aside from geographic data, and provide an alternative approach to pattern interpretation. These details are explained in the next chapter.

Chapter 3

Methodology

This chapter provides a discussion of the proposed framework for understanding movement patterns, which was initially introduced in [66]. The framework consists of three major phases, namely, pattern discovery, semantic annotation and pattern analysis. Since these phases correspond to the KDD phases, the figure on the KDD process is reiterated here as shown in Figure 3.1. It is important to recall that KDD aims to extract useful knowledge from the input data and this is realized through its three main phases, which are preprocessing, data mining, and interpretation/evaluation.

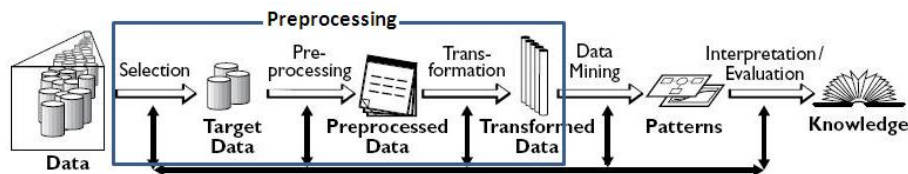


Figure 3.1: Overview of the steps constituting the KDD process. (Based on [29])

The phases of the proposed framework correspond to the last two phases of KDD, as illustrated in Figure 3.2. Specifically, pattern discovery corresponds to the data mining phase of KDD while semantic annotation and pattern analysis correspond to the interpretation/evaluation phase.

The pattern discovery phase involves extracting patterns inherent in the considered dataset using a specific mining algorithm selected by the user. Once the patterns are found along with their properties, there is a need to relate them with the context, which is essential for extracting meanings from the discovered patterns. To do this, the trajectories and the patterns are linked to their corresponding semantic attributes, which represent the context or at least part of the context, during the semantic annotation phase. Finally, in the pattern analysis phase, the annotated trajectories and patterns are analyzed using data mining techniques in order to aid the user in interpreting the discovered patterns.

The framework as a whole allows domain experts to interpret patterns by deducing the possible interactions that might have taken place. This is very important because it can support the understanding of social behavior among the observed entities.

The framework is also general and applicable to different types of movement patterns. In order to test its applicability and effectiveness, we have instantiated the framework to interpreting moving flock patterns, which is a specific type of flock involving members that

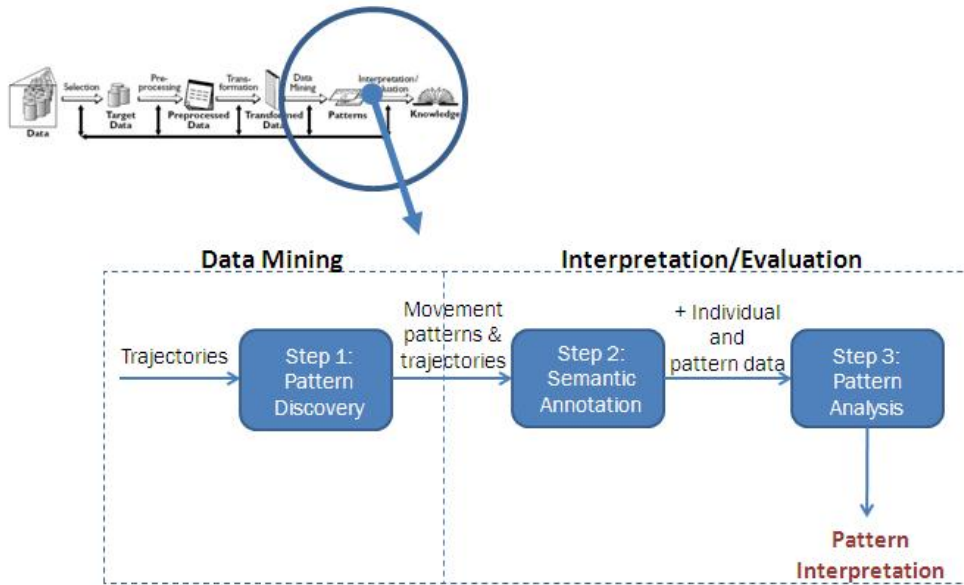


Figure 3.2: Proposed framework for interpretation of movement patterns.

move together from one place to another. We have found such patterns interesting since flock patterns, in general, contain implicit information about the members' interaction with other entities. Moreover, analysis of moving flocks can support the understanding of collective movement behavior as opposed to an aggregated convergence behavior with stationary flocks. Thus, the discussion of the different phases of the proposed framework includes details that are specific to processing flock patterns.

3.1 Pattern Discovery Phase

The purpose of the pattern discovery phase is to extract movement patterns from the given dataset. These patterns contain implicit information that can lead to understanding the movement behavior of the observed entities. The discussion on the remaining part of this section focuses on moving flock patterns.

Though there are already existing flock discovery algorithms described in literature, some of them are mainly theoretical and have not been implemented. Those that were implemented were either not accessible or do not fit our need. For instance, initial works [51, 38] in this field require an additional mechanism for extracting flocks that last for more than one time instance. In addition, existing flock algorithms extract both moving and stationary flocks, without distinguishing between them. This distinction is important since these are two different types of patterns and consequently, they have different semantics. For these reasons, we have developed a data mining algorithm that focuses on the discovery of moving flock patterns.

3.1.1 Moving Flock Definition

Prior to giving the definition of moving flock patterns, we will first define some related concepts. Recall from Section 2.2.2 that a trajectory can be defined as a sequence of

(x, y, t) -tuples describing its position over consecutive time instances in ascending order. It can also be defined as a sequence of line segments that connect every pair of adjacent points (x, y, t) . More formally, trajectory T can be defined as follows: $T = \langle LS_1, \dots, LS_{n-1} \rangle$ where each LS_i is the line segment that connects points (x_i, y_i, t_i) and $(x_{i+1}, y_{i+1}, t_{i+1})$, and $1 \leq i \leq n - 1$. Therefore, if there are n points in a trajectory, this implies that it contains $n - 1$ line segments. The number of line segments in an entity's trajectory may vary with that of another entity's trajectory since different entities move in different ways and thus, the trajectories they produce vary in length as well.

A trajectory, in turn, is also composed of sub-trajectories. As its name implies, a sub-trajectory is a smaller sequence of (x, y, t) -tuples contained in the original trajectory. More formally, we define a sub-trajectory of T over a time interval I , denoted by T_I , as $T_I = \langle LS_s, LS_{s+1}, \dots, LS_{e-1} \rangle$ where $I = [t_s, t_e]$ (i.e., T_I is composed of segments in T that correspond to the time interval I). This concept is useful in the discussion of moving flocks since entities may not always flock during the entire duration of their movement. Consequently, the event of flocking only occurs on a sub-trajectory of the flock members and not on entire trajectories.

We will now define the spatial extent of a flock, which is a key notion that we use for distinguishing moving from stationary flocks. It is a measure of the amount of space covered by a group of entities during their time of flocking as represented by their sub-trajectories. More formally, given a flock F (as defined by Benkert et al in [15]) in a time interval I , its spatial extent, $ext(F, I)$, is defined as $ext(F, I) = \max\{l, w\}$ where l and w are the length and the width of the minimum bounding rectangle (MBR) of the set of sub-trajectories belonging to the flock. We selected MBR to compute the spatial extent due to its simplicity and efficiency, but other computational techniques can be selected as well. The same notion of extent can also be easily applied to a single trajectory and we call this the trajectory extent.

We are now ready to give a formal definition of a moving flock pattern.

Moving Flock Definition: Given a set of n trajectories consisting of line segments (i.e., sub-trajectories) that can vary in number for different trajectories, an (m, k, r) -moving flock F_M in a time interval $I = [t_i, t_j]$, where $j - i + 1 \geq k$, consists of at least m objects such that for every discrete time step $t_l \in I$, there is a disk of radius r that contains all the m objects and the spatial extent $ext(F, I) \geq r$. Simply put, it states that a moving flock is a group of entities that consists of a minimum number of members, and these entities move from one location to another in a manner that they remain spatially close over a minimum time interval.

Compared to existing flock definitions, our definition distinguishes moving flocks from stationary flocks, which are group of entities that stay closely together in only one location during the time interval of flocking. It is important to distinguish between the two since users may only be interested in moving flocks depending on the application context. For example, in the movements of people in the park, the park manager may be interested in how a group of visitors move together in certain parts of the park and differentiate this from a group of visitors who meet in a certain location without moving together to other interesting locations.

In distinguishing between moving and stationary flock, we chose to compare the flock's spatial extent with r . Since r is the measure used to determine spatial closeness among a group of objects (i.e., the distances among these objects are very small if they are at most equal to r), it can also be used as a coverage measure of the spatial extent. If a flock

covers an extent smaller than r , this means that over several time steps in time interval I , the flock is “moving” at short distances (i.e., “moving” to positions that are very close). Due to GPS error, movement at short distances can be interpreted as being stationary. Thus, we chose r to filter out stationary flocks. Instead of using r , a factor of r can be used as well. In order to avoid an additional parameter, we chose this factor to be always equal to 1.

It is worth noting that the definition focuses on fixed moving flocks, which consist of the same members over the considered time interval. As opposed to flocks with varying members, it emphasizes the importance of each member’s identity, which is interesting in the context of pedestrian movement, especially for the purpose of further analyzing discovered flock patterns.

Finally, our algorithm provides users with a certain measure of flexibility through the following set of parameters since they may be interested in different types of moving flocks depending on the application domain:

- *min_points* - the minimum number of objects that are members of a moving flock. It is equivalent to m in the definition.
- *min_time_slices* - the minimum number of consecutive time instances for which the flock members are close together. Discovered patterns with shorter time instances are not considered as flocks since the members are close for only a short period of time. It is equivalent to k in the definition.
- *radius* - defines the closeness of members at some time instance. The flock members are close at a time instance if they belong to some disk with a specified radius, and with one of the members located at the center. It is equivalent to r in the definition.
- *synchronization_rate* - the fixed time rate (expressed in seconds) at which observed points (e.g. GPS recordings) are sampled for each flock member. In other words, it is the temporal gap between two sampled points. This parameter is needed for the synchronization step of the flock discovery algorithm and will be described further in the next section.

These parameters are used to measure the spatio-temporal coherence that should exist among members of the same flock. By spatio-temporal coherence, we refer to the consistency of having spatial closeness among members over the period of flocking.

3.1.2 The Moving Flock Discovery Algorithm

We propose a four-step approach for extracting moving flocks from an input dataset. It includes synchronization, spatial neighbor computation, membership persistence analysis, and pruning. We shall start with a brief description of the preprocessing task performed prior to these steps.

Preprocessing Due to the presence of inaccuracies and noises in movement datasets, preprocessing is normally performed prior to performing mining tasks. We used a separate tool called M-Atlas [85] (previously known as Daedalus [68]), which provides support for moving entity data mining query language, for this task. The tool allows the user to

restrict certain properties among points belonging to the same moving object. Some of these properties are the number of points, the time gap, the spatial distance, and the speed. More specifically, the trajectory of each object are cut further into shorter trajectories or noises are removed from it in the case that any property between two consecutive points of an original trajectory is larger than its user-defined threshold.

The next paragraphs describe the steps of the moving flock discovery algorithm.

Synchronization Step

Synchronization of the recorded points in the raw tracking dataset is necessary since (x, y, t) observations are usually recorded at random time instances. In order to compare the closeness of points for corresponding time intervals, trajectory points belonging to different entities should be sampled at regular time steps. This synchronization can be performed by sampling (x, y, t) points at the rate provided by the user, which we refer to as the *synchronization_rate* parameter. For example, a *synchronization_rate* set to 300 means that points are sampled at the rate of 300 seconds. Linear interpolation is performed to approximate (x, y, t) data that are not found in the raw dataset. M-Atlas is utilized to perform sampling of the points in each trajectory at regular time steps. Figure 3.3 shows how points are sampled at regular time steps. The circles in the figure are the recorded points in the raw dataset while the squares refer to the sampled points computed using linear interpolation. The dotted lines refer to the regular time instances at which the points are sampled. The figure summarizes the synchronization step, which basically translates raw points into points that are sampled at a regular rate.

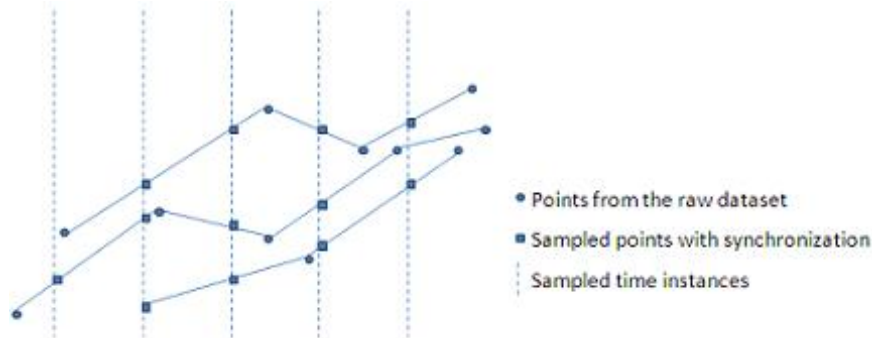


Figure 3.3: Sampling of points at regular time interval.

After the completion of this step, the algorithm would have a nice dataset having trajectories that contain (x, y, t) -tuples sampled at the specified synchronization rate.

Spatial Neighbor Computation Step

After performing synchronization as a preprocessing step, the existence of spatial coherence at each synchronized time instance is checked during the spatial neighbor computation step. Every trajectory is considered as the base trajectory in order to determine the entities that are spatially close to it for its sampled time instances. Considering a specific base trajectory at a time, its spatial neighbors for each time instance are computed by drawing a disk with the base trajectory's position as its center. Moving entities with points

lying within this disk are considered as the base trajectory’s neighbors for the considered time instance. Thus, after the completion of this step for a base trajectory, there would be t computed disks corresponding to the t time instances of the base trajectory. Each of these disks represents the spatial neighbors of the base trajectory for each time instance.

Figure 3.4 provides an example of the described step. In the example, the spatial neighbors of the considered base trajectory are computed for each sampled time instances of the base. The spatial neighbor of the base trajectory at the first time instance consists of *trajectory B*, while it includes *trajectories A* and *B* from the second to the fifth time instances. The base trajectory has no spatial neighbors at the last time instance.

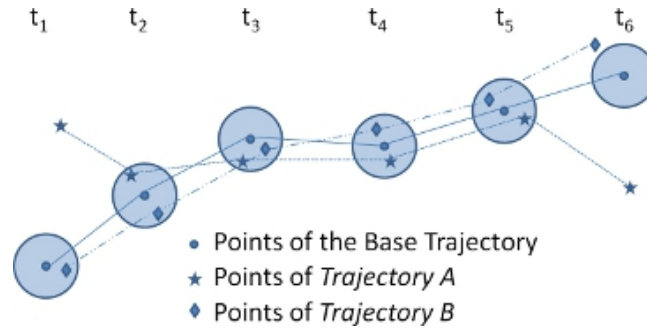


Figure 3.4: Computation of the base trajectory’s neighbors at each time step.

The *radius* parameter, which is the measure of spatial closeness, is fixed to a single value for each execution run of the algorithm. Furthermore, the computed disks for different time instances may also spatially overlap which means that the moving entities are stationary or barely moving during such time periods.

Membership Persistence Analysis Step

Each disk generated in the spatial neighbor computation step represents a group of pedestrians that are close to the base object. This group, including the base, can be seen as a ‘basic flock’, which refers to a candidate flock that is known to exist for one time instance.

The completion of the spatial neighbor computation step, however, only covers the check for spatial coherence among the candidate flock members. There is also a need for executing the membership persistence analysis step, which checks the persistence of the ‘basic flocks’ over time. More specifically, it involves checking whether members of a ‘basic flock’ are also found in other ‘basic flocks’ of adjacent time instances. Doing so verifies whether members at a specific time instance continue to be close to each other for other adjacent time instances.

In this step, each base trajectory with its corresponding disks is considered one at a time. Disks in adjacent time instances are merged if the number of common members in both disks is at least equal to the user-defined *min_points* threshold. This merging process is performed in a recursive manner. From a pair of ‘basic flocks’, a ‘composite flock’ (i.e., a candidate flock occurring for more than one time instance) is formed if the members persist over the two adjacent time instances. Once all ‘basic flocks’ are processed, each ‘composite flock’ is treated as a ‘basic flock’, causing ‘composite flocks’ that last for two time instances each to be merged with other ‘composite flocks’ in adjacent time instances

if their members persist. This merging process continues until longest duration flocks are found.

Figure 3.5 illustrates the order of merging ‘basic flock’ patterns in order to form the longest duration flock patterns. The order is indicated by the number found in each bracket. A bracket represents the check performed between a pair of disks belonging to adjacent time slices. In the best case scenario wherein every disks of the base trajectory contain the same members satisfying the minimum points requirement (i.e., m threshold in the moving flock definition), all disks in the base trajectory will eventually be merged into a single disk that persists from the first time instance to the last time instance of the base.

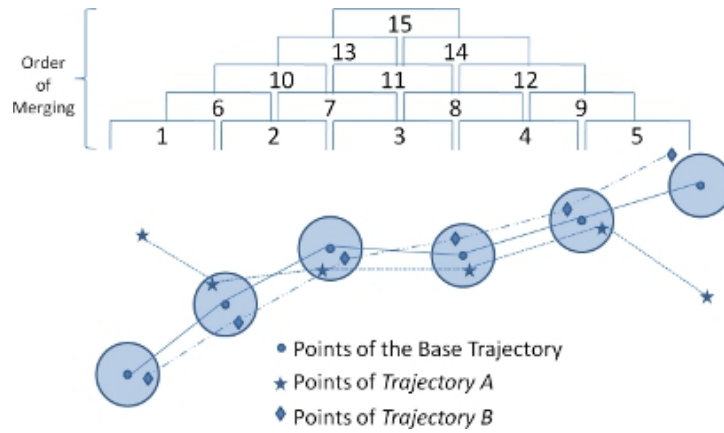


Figure 3.5: Order of merging candidate flocks at each time step into flocks that persist over a certain time duration.

Pruning Step

Once candidate moving flocks are discovered, the final step is to remove short-duration, stationary and redundant flocks during the pruning step. This corresponds to three levels of pruning. The first and simplest step is to discard patterns with time instances shorter than the user-defined threshold *min_time_slice*. The second level involves pruning out redundant patterns generated when using related base trajectories. Finally, stationary flocks are also filtered out.

Redundant patterns refer to a set of patterns having entities and duration that are almost the same (i.e., majority of the patterns’ entities are the same and a large interval of their duration is also the same). Such patterns normally results from using base trajectories whose spatial positions are very similar during the time of flocking. Hence, generating flock patterns that are very similar as well. The research assumption behind the pruning technique performed for removing redundant patterns stems from the notion that a member can only belong to one flock at a time. Thus, if a member already belongs to a flock for some time instances, it is no longer considered as a base for those time instances so as not to further produce flocks that include members already belonging to some other flock. This step improves the running time of the algorithm besides reducing redundancy in the results.

A problem with this approach, however, is the possibility of losing some moving flock

patterns since some sub-trajectories are disregarded as bases. Using these disqualified bases can lead to discovery of flocks that are quite similar to previously discovered flocks in the sense that their time duration and members are overlapping. If it is important for the user to retain such patterns, the condition for disregarding can be relaxed (ex: disregard bases only when the flock that can be discovered from them is quite similar to previously discovered flocks) or removed altogether, but this leads to higher running time and more redundancy in the results.

Meanwhile, the stationary flock patterns are pruned when the extent of each discovered flock has been computed. Candidate patterns with a short extent (i.e. having an extent smaller than the user-specified radius) are considered as stationary flock patterns and are thus, disregarded. The research assumption behind this step is that flocks with members who only cover a short extent are most probably stopping together in a specific location. For example, flock patterns with members who are spatially close for a long time interval but only stays in one location are pruned out. On the other hand, the approximation approach still considers moving flock patterns with members who stop within its flocking time interval if the whole extent is long enough. For instance, a moving flock may stop together for some time before moving together.

As mentioned earlier, the flock's spatial extent is computed by finding the Minimum Bounding Rectangle (MBR) for each sub-trajectory included in the flock considered. This is computed by first finding the length and width of each sub-trajectory's MBR. The maximum of the length and the width is the sub-trajectory's extent. The minimum of these extents is the flock's spatial extent. Figure 3.6 provides an example. The MBR is computed for each sub-trajectory of flock members during the flocking duration. From the MBRs, the length and width can be extracted. Since the longer part of *Sub-traj1* segment is the width, *width1* is assigned as its extent. Same is true for *Sub-traj2*. Finally, the flock extent is the minimum of all sub-trajectory extents, which is *width2* in this case.

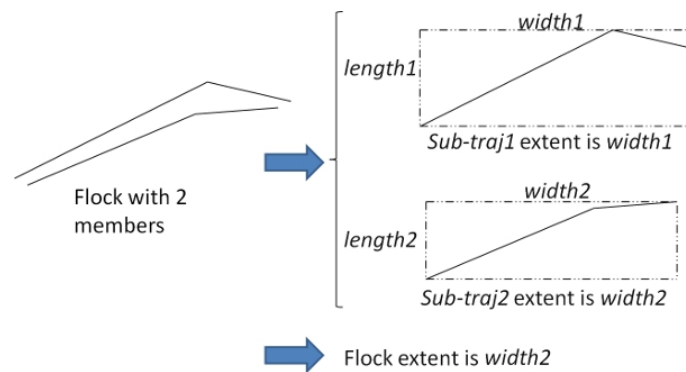


Figure 3.6: Example of computing the flock extent for a flock consisting of two members.

Applying the four steps of the moving flock discovery algorithm on a given tracking dataset produces a set of moving flock patterns that satisfy the constraints specified in our definition. The discovered patterns exclude stationary flocks and the number of redundant patterns is reduced. It may also be interesting to retain the stationary flocks and to classify flocks as either moving and stationary, especially when the discovered flocks are few in numbers.

It is worth noting that due to the first-come-first-serve nature of this phase, especially

in the filtering of base trajectories to avoid redundant patterns, entities found in the first part of the dataset are more likely to be utilized as base trajectories compared to those found in the latter part. As a consequence, the order of the entities in the input may cause slight differences in the obtained results. In order to understand the impact of the entities' ordering on the flock results, we tested the algorithm on the same dataset having varying versions that only differ in the entities' ordering, and compared the obtained results. We have found that the order indeed may vary the flock results. However, the difference in the results are mostly minor (i.e., members and time duration are still largely overlapping in most cases). The details of this experiment can be found in Chapter 4.

The worst-case time complexity of the algorithm is $O(n^2t + nt^2)$, where n is the total number of trajectories and t is the maximum trajectory length in terms of time. The first term n^2t comes from the spatial neighborhood computation and pruning steps while the second term nt^2 is due to the membership persistence analysis step. Since the synchronization step is only executed once, its time complexity contribution is dominated by the other steps.

The next two sections cover issues related to the moving flock discovery algorithm. The first issue found in Section 3.1.3 is related to its parameters while the second issue found in Section 3.1.4 is on its validation.

3.1.3 Selection of the *Radius* Parameter

As mentioned earlier, the algorithm provides the user with flexibility through four user-defined parameters, whose value depends on several context-dependent factors, such as the topology of the area and the type of information that the analyst would like to obtain. This section aims to provide a general guideline for finding an appropriate value for this set of parameters although finding the best parameters for a specific dataset is still an open issue. A trial-and-error approach along with statistical computations can help in finding a suitable set of parameters.

We provide general rules of the thumb for finding an appropriate value for the *min_points*, *min_time_slices*, and *synchronization_rate* parameters. As for the *radius* parameter, we extend a technique employed in DBSCAN [28] to suggest a good value for it.

The following are general rules of the thumb. *min_points* depends on whether the user is interested in flocking behavior of couples, of a small group, or of a big group. On the other hand, *min_time_slices* depends on whether the user is interested in short or long periods of flocking. A good value for a short period of flocking would be at least 3. While 2 is also a valid value, it may be too small since this means that the entities were close for only 2 consecutive time instances. Meanwhile, the value of *synchronization_rate* can be guided by the sampling rate used in the raw dataset and by the types of flocks (ex: flocking on pedestrian lanes, on paths, etc.) that the user is interested in. If the original sampling rate is around 5s, then *synchronization_rate* should not vary too much from this value. A value of around 1 minute can be a good value for the synchronization while a value of 2 minutes or more may be too large.

Perhaps, the most challenging parameter to define is the *radius* parameter. The user can be guided by intuition in deciding the appropriate distance between two objects that are considered spatially close. For example, if the user is interested in finding flocks that occur in pedestrian lanes, the radius would depend on the length and width of these lanes. However, at the same time, the user should also consider the uncertainty introduced by

the location technology used in collecting the data. Thus, there should be provisions for this uncertainty in the radius value.

To further guide the user in determining the *radius* parameter, we adapted the technique used in DBSCAN in determining its *Eps* parameter. *Eps* serves a purpose that is comparable to that of the radius. It is used to separate core and neighboring points from noise points. While DBSCAN deals with 2D data represented by (x, y) pairs, the flock algorithm deals with 3D data represented by (x, y, t) tuples that are connected through object or trajectory IDs. Thus, necessary adjustments to their technique is necessary to accommodate the time component.

For each trajectory in the dataset, its distance from its k -th nearest neighbor is computed. We will refer to this distance as the k -th distance, and k is equal to $min_points - 1$. It is important to note that this distance is computed with the consideration of time. That is, the x and y components of a pair of trajectories are only compared if their occurrence overlaps in time. The k -th nearest neighbor is obtained by considering an trajectory's distance from all other trajectories over all time instances found in the dataset. Once the k -th distance is computed for each trajectory, these distances are sorted in increasing order and plotted as a line graph. The point in which there is a sudden increase in the k -th distance can be a good value for the *radius* parameter of the flock algorithm. The research intuition behind this approach is that the point of sudden increase separates flock members from non-flock members since the radius has to be increased to a very large value in order to find a few more flocks at this point.

Figure 3.7 gives a good example of such plot since there is a clear point of sudden increase in the distance, which is approximated at 300 meters. It shows the trajectories' distance from their 3rd nearest neighbor and suggests that 300 meters can be a good *radius* value for finding flocks in the dataset. Depending on the dataset and the choice of k , this point of sudden increase may not exist.

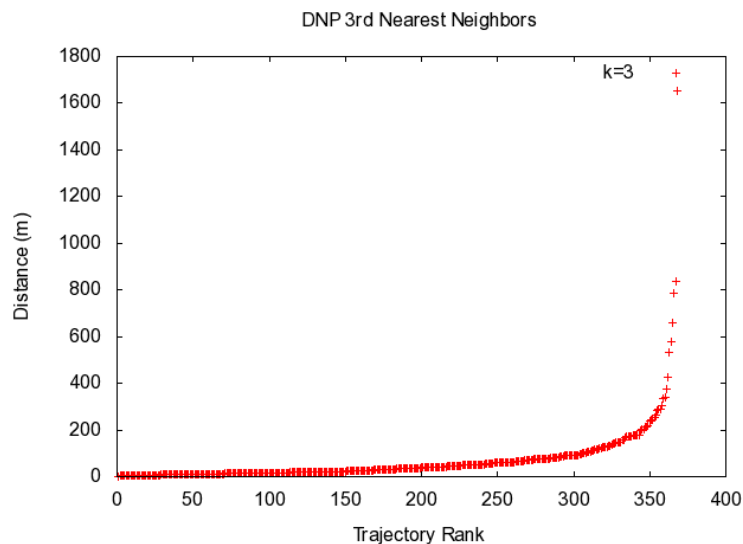


Figure 3.7: Sample Plot of k -th Distances.

Besides introducing the application of this technique for finding a suitable radius, we

have also run the flock algorithm several times on the same dataset with the same set of parameters except for varying radius values. The purpose of this experiment is to understand how changes in the radius affects the flock results. The experiment and its results are discussed in more details in Chapter 4.

3.1.4 Validation of the Flock Discovery Algorithm

Aside from developing an algorithm that extracts moving flocks from an input dataset, we have also proposed the use of two combined methods for validating the algorithm. The first method mainly relies on the visualization of the extracted flocks while the second method is founded on the null hypothesis principle [41]. The application of the first method verifies that the obtained results are indeed flock patterns while the second method provides the assurance that the algorithm is working properly in the sense that the results extracted by this algorithm are indeed moving flocks inherent in the dataset and not simply extracted by chance.

Since flocking is often associated with an image of birds or other types of entities that are moving closely together on a certain route over a specific time duration, the most natural way of checking for the occurrence of flocks is through the sense of sight. For this reason, we have initially validated the algorithm by plotting the trajectories of flock members over the duration of flocking, which is done for a sample of the obtained flocks. We chose the top-ranking flocks and the lowest-ranking flocks, in terms of their flock extent, as the samples. Then, for each flock, the trajectories of its flock members over the period of flocking can be plotted using existing visualization tools. By looking at each plot, the independent domain expert can assess that the obtained moving flock is correct if the trajectories made by its flock members are spatially close, and the length of the trajectories are long enough to be able to say that they are not simply staying in one location. Figure 3.8 gives two plots, the left plot being a moving flock example and the right being a non-example. The plot of the moving flock example shows how the entities moved closely together over the flocking duration. On the contrary, the non-example shows that the entities were moving in different directions. It is actually a zoom-in on a stationary flock, and was classified as a flock in the general sense since these sub-trajectories can be completely contained in the disk defining the spatial neighbors of the base trajectory.

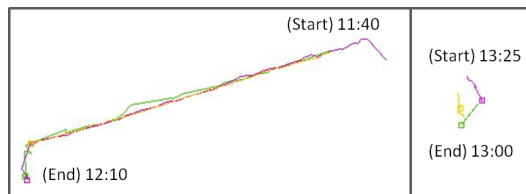


Figure 3.8: The left plot is a moving flock example while the right plot is a non-example.

After visually inspecting the results and observing that the obtained flocks are indeed moving flocks, the next step is to assure the user that these flocks are inherent in the input dataset. This can be achieved by applying the null hypothesis wherein we initially assume that the extracted flocks are obtained by chance. In order to disprove this hypothesis, a randomization step is performed on the original dataset several times, each time producing a different randomized dataset (i.e., a distorted version of the original dataset).

Afterwards, the flock discovery algorithm is executed having each randomized dataset as input. The results obtained from applying the algorithm to the original dataset and to the randomized datasets are then compared. Obtaining different flock results from different input datasets demonstrates that the algorithm produces results that are not obtained by mere chance but on the contrary, are based on the nature and the characteristics of the input dataset. Further evaluation of the discovered flocks is still needed in order to assess the quality of the discovered patterns.

We propose the use of the following techniques for dataset randomization:

Randomization by Markov Chain One technique for randomizing the synchronized input dataset is to build a Markov chain based on the dataset distribution. This model assumes that only the current state affects the next state while past states and future states are irrelevant.

The initial step involves building the model by computing the probabilities of transitions between the x - y - coordinate values. A transition from (x_i, y_i) to (x_{i+1}, y_{i+1}) exists if they belong to the same moving entity, and time instance $i + 1$ immediately follows time instance i . Since there is a large number of varying x - and y -values in the datasets, the x - and y -values are grouped into grids. Then, the probabilities of transitions between these grids is computed. Analogous to an existing transition between a pair of x - and y -values, a transition from $grid_i$ to $grid_{i+1}$ exists if there exists a transition from (x_i, y_i) to (x_{i+1}, y_{i+1}) such that (x_i, y_i) belongs to $grid_i$ and (x_{i+1}, y_{i+1}) belongs to $grid_{i+1}$.

This randomization technique modifies the x and y values based on the described Markov chain while retaining the original entity ID and the original time values found in the synchronized dataset. The (x, y) -pair for the first time instance of an entity is a random value biased towards the most probable initial (x, y) -pairs found in the dataset. The succeeding (x, y) -pairs are determined based on the immediately preceding (x, y) -pair and the Markov chain. More specifically, the grid in which the current (x, y) -pair belongs to is determined before computing the next grid using the Markov chain. Once the next grid is computed, the next (x, y) -pair can be computed as a random value limited by the bounds of this grid. In the case that there is no next probable grid, a new trajectory is started by randomly picking a most probable initial (x, y) -pair and continuing in a manner as described before.

Randomization based on Geographical Coordinate Uncertainties It is known that the collected observation points contain inaccuracies due to the limitation of current location technologies. Considering this uncertainty, we propose another randomization technique that modifies the x -, and y -values of the dataset by using values of radius as a measure of uncertainty. The research assumption here is that a larger radius of uncertainty would produce a dataset that is very different from the original while a smaller radius would produce a dataset that is quite similar to the original. Hence, a dataset randomized with a large radius value should contain inherently flocks that are very different from that of the original while a dataset randomized with a small radius value should contain flocks that are very similar to those of the original.

As with the previous technique, the original IDs and the original time values are retained and only the (x, y) -pairs were randomized. A user-defined radius, which represents the uncertainty in the coordinate values, is subtracted and added from an existing

(x, y) -pair. The obtained difference and sum determines the range used for computing the corresponding randomized value.

Aside from the presented validation techniques, cluster validation measures can be instantiated and extended to flocks in order to assess the quality of flock results.

3.2 Semantic Annotation Phase

The previously described phase of the framework, the pattern discovery phase, is mainly concerned with extracting flock patterns from the considered dataset. These extracted patterns, however, only explicitly provides information about the positions of flocking, the members who flocked together, the duration of their flocking and possibly the speed of flocking depending on the data mining algorithm used. The reason for flocking and the specific interactions involved during flocking are still hidden from the user. This is addressed in the last two phases of the framework, which are the semantic annotation and the pattern analysis phases.

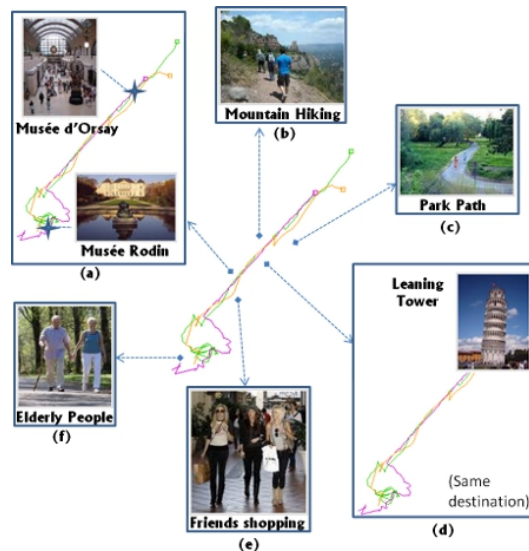


Figure 3.9: A specific instance of flock pattern that can be interpreted in several possible ways depending on the movement context.

Before flock analysis can be performed, there is a need to semantically annotate the extracted flock patterns and the flock members in order to set up the concepts that can describe the context in which the movement behavior occurred. This plays a primordial role in the interpretation of a flock pattern depending on the context. Recall the example given in Chapter 1. It is shown again in Figure 3.9 for ease of reading. As mentioned previously, this example illustrates the importance of considering the movement context since varying contexts result in varying interpretations. In order to take the context into account, semantic annotation is necessary for incorporating semantic information that describe the surrounding context.

During the semantic annotation phase, the trajectories and the discovered flock patterns are augmented with semantic information that contributes to the description of the context in which the movements occur.

3.2.1 Semantic Attribute Source

Before semantic annotation can take place, it is necessary to first identify semantic sources to be used for annotation. This subsection provides an overview of such sources, which include questionnaires/survey, thematic and topographic maps, information on points of interest, census data, and others.

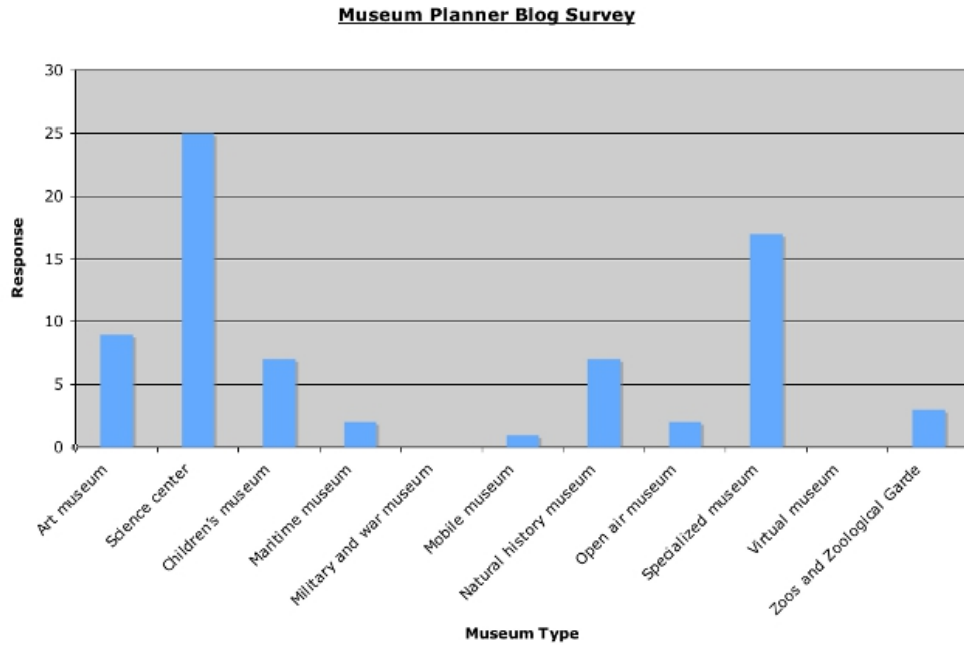


Figure 3.10: Users' responses to their interest in museum planning. [63]

Questionnaires/Surveys Questionnaires and surveys are rich sources of semantic information. A questionnaire has been defined in Merriam-Webster [59] as “a set of questions for obtaining statistically useful or personal information from individuals” or “a survey made by the use of a questionnaire”. For this reason, questionnaires and surveys are used interchangeably in this thesis.

In our experiments, the main source of the semantic information used to annotate the discovered flock patterns is the collected responses of pedestrians to conducted surveys. The DNP dataset, for instance, do not only contain the (x, y, t) observation points but survey responses of each entity as well. A sample question in the survey is as follows: *Are you in Dwingelderveld for vacation? [Yes/No]*.

Another example is in the context of museum planning as described in [63]. Sample questions in the conducted survey are: (a) What type of museum planning are you most interested in? (b) Please select the description that best describes your profession/position. The responses to these questions are summarized in Figure 3.10 and 3.11, respectively. An analysis of the combined results can provide interesting insights to the relation between people's profession and their interest in museum planning. This example also demonstrates how surveys can now be conducted online instead of being solely paper-based.

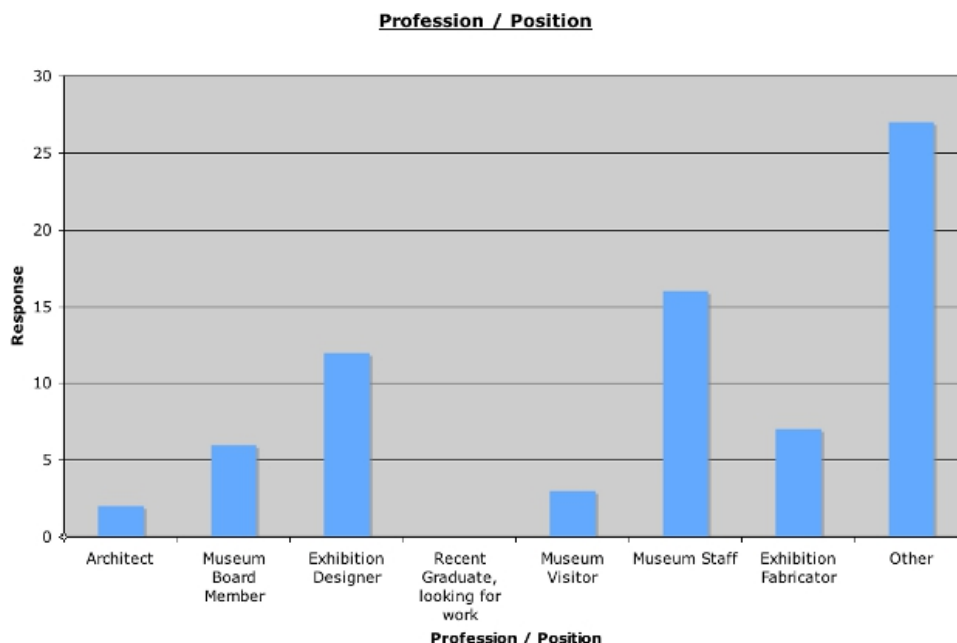


Figure 3.11: Users' responses to their profession/position. [63]

Thematic and Topographic Maps Another source of semantic information are thematic and topographic maps. A thematic map is a special type of map that emphasizes certain theme or special topic such as the average distribution of rainfall or the distribution of religious sectors in an area [33]. An example of which is a world climate map shown in Figure 3.12.

Meanwhile, a topographic map is a specific type of thematic map that emphasizes the terrain features of the area [33]. A topographic map of Georgia is shown in Figure 3.13, which emphasizes its relief features.

The information contained in both thematic and topographic maps are useful in describing semantics related to the geographic location, which are not restricted to geographical properties. For example, the movement of entities in an area in which the weather is mostly sunny would be different from those moving in a rainy area. The religion or the form of government in an area may also affect people's movement.

Information on Points of Interest POIs (Points of Interest) refer to interesting locations or attractions in a specific area. They are usually used in online maps such as those provided by Google, Yahoo, and OpenStreetMap. Figure 3.14 illustrates a map of a Pisa sub-area, which is obtained from OpenStreetMap. It contains different examples of POIs like churches, a bank, restaurants, pizza shops, and cafés. With the advancements in location technologies, spatial coordinates of POIs along with their corresponding description has now been collected and made available in sites such as <http://www.gps-data-team.com/> and <http://www.downloadpoi.com/>. Such collections can be exploited for semantic annotation in order to link the spatial positions of moving entities with their description. For instance, knowing that a person visited the leaning tower of Pisa is more meaningful than stating that the person visited the latitude and longitude

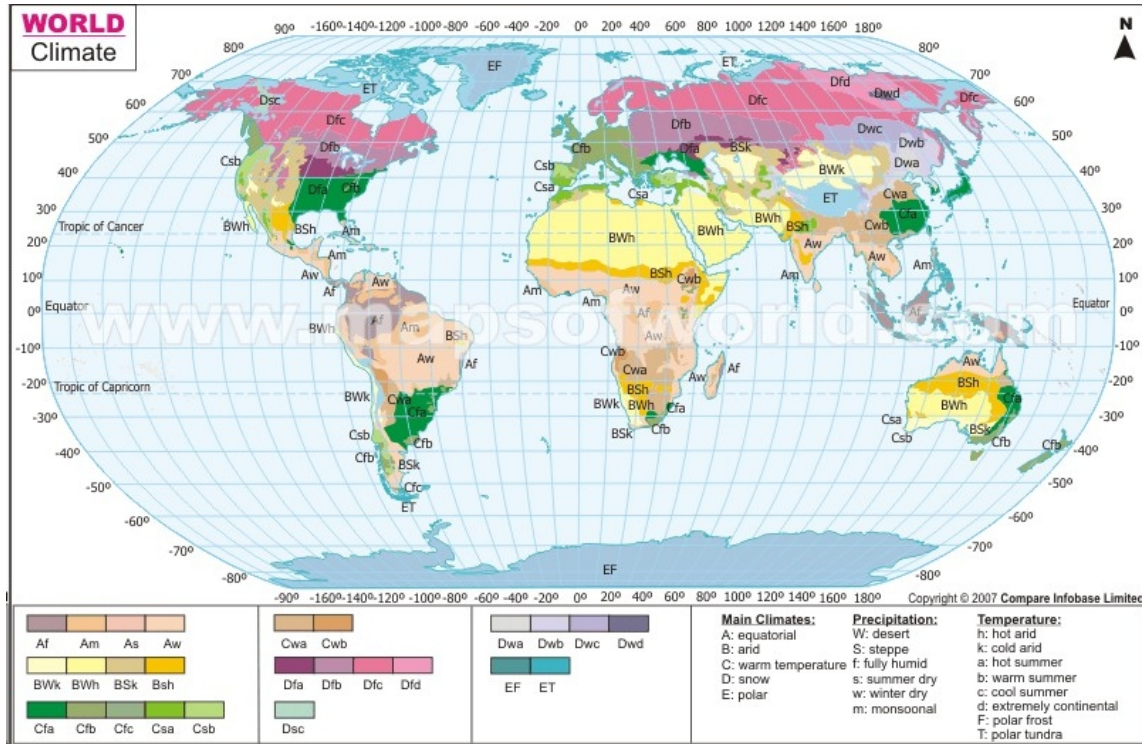


Figure 3.12: A world climate map. [93]

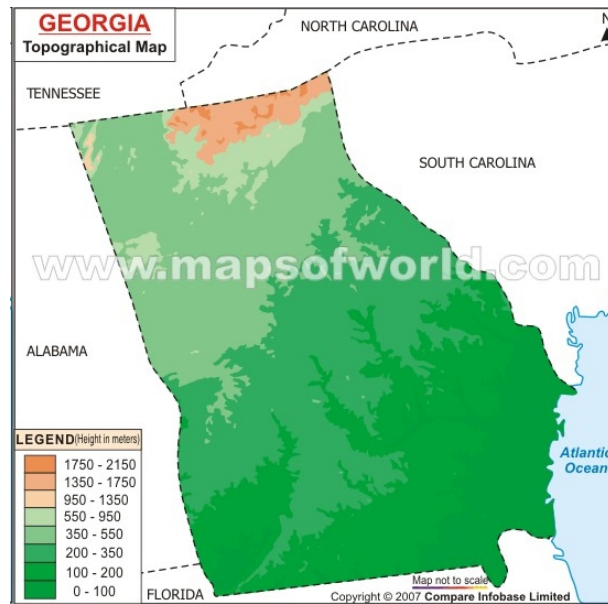


Figure 3.13: A topographic map of Georgia. [93]

coordinate (43.72313,10.396768).



Figure 3.14: Some POIs in Pisa.



Census Data A census refers to a systematic procedure of gathering and recording information about the members of a given population [19]. The recorded data, referred to as census data, can provide statistical information that can be useful for semantic annotation. Figure 3.15 provides a fragment of statistical information contained in the 2006 Census Data for Central Northern Sydney. It contains both individual and household characteristics, which can affect movement behavior of the population. Age, country of birth, main language spoken at home, religious affiliation, marital status, and occupation are some examples of individual characteristics included in the census data. On the other hand, dwelling characteristics, tenure type, household composition, and landlord type are examples of household characteristics. Basically, the statistical information describes the distribution of the population, based on the individual and household characteristics, in Central Northern Sydney. For instance, 48.7% of the population excluding overseas visitors are male.

Others There are still many other possible sources of semantic information, such as environmental reports and domain expert knowledge. Note that compared to the previously discussed sources, these are less structured. In fact, the knowledge of domain expert may not be structured at all if it is not documented.

Since the number of thematic and geometrical attributes related to a pattern and/or a moving individual can be large, attribute selection can be a difficult task. A simple random choice is possible, but this can lead to meaningless results when correlation analysis is performed. Therefore, a specific contribution of this thesis is the application of a criteria set for the purpose of selecting meaningful attributes among all the available attributes, as explained in the next section.

[Person Characteristics](#) | [Age](#) | [Selected Characteristics](#) | [Country of Birth](#) | [Main Language Spoken at Home](#) | [Religious Affiliation](#) | [Marital Status](#) | [Labour Force](#) | [Occupation](#) | [Industry of Employment](#) | [Income](#) | [Family Characteristics](#) | [Dwelling Characteristics](#) | [Dwellings Characteristics Occupied Private Dwellings](#) | [Tenure Type](#) | [Household Composition](#) | [Landlord Type](#)

PERSON CHARACTERISTICS (Place of usual residence)

PERSON CHARACTERISTICS 	Selected Region	% of total persons in Region	Australia 	% of total persons in Australia
Total persons (excluding overseas visitors)	4,119,190	-	19,855,288	-
Males	2,028,729	49.3%	9,799,252	49.4%
Females	2,090,461	50.7%	10,056,036	50.6%
Indigenous persons (comprises Aboriginal and Torres Strait Islander)	43,518	1.1%	455,031	2.3%

In the 2006 Census (held on 8th August 2006), there were 4,119,190 persons usually resident in Sydney (Statistical Division): 49.3% were males and 50.7% were females. Of the total population in Sydney (Statistical Division) 1.1% were Indigenous persons, compared with 2.3% Indigenous persons in Australia.

[Back to top](#)


AGE 	Selected Region	% of total persons in Region	Australia	% of total persons in Australia
Age groups:				
0-4 years	270,814	6.6%	1,260,405	6.3%
5-14 years	534,214	13.0%	2,676,807	13.5%
15-24 years	569,896	13.8%	2,704,276	13.6%
25-54 years	1,816,105	44.1%	8,376,751	42.2%
55-64 years	422,182	10.2%	2,192,675	11.0%
65 years and over	505,979	12.3%	2,644,374	13.3%

Figure 3.15: Fragment of 2006 Census Data for Central Northern Sydney. [9]

3.2.2 A Guideline for Semantic Attribute Selection

Aside from identifying the sources of semantic information, it is also important to understand which of these information are relevant to the semantic annotation step. Therefore, we propose the use of a guideline based on Wood and Galton's taxonomy for collective phenomena presented in [92]. Recall that this work was described earlier in Section 2.5.2 and it proposes a taxonomy of collectives based on a set of criteria, which includes membership, location, coherence, roles, and depth. Although an overview of these criteria was already given earlier, we provide a more detailed discussion in this section in order to explain how these criteria can be used for semantic attribute selection. Before this, it is important to first understand the motivation for adapting this work over other related works on ontologies.

Motivation To start the discussion on the reasons for choosing Wood and Galton's work over others, we must first emphasize the consistency of our perception of collectives with their insights as described in [92]. We both recognize a collective as a group of entities that is seen as a whole and thus, it exhibits properties that are distinct from the individual properties of its members. For example, the age of individual members in a collective is different from the age characteristic of the entire collective. Meanwhile, other related works on collective ontology, such as [17], have a different perception on collectives. Specifically, [17] restricts the definition of collectives to a group of entities that are unified

by a plan.

A main motivation for considering the set of criteria for classifying collectives is the strong link between movement patterns and collectives. Since many patterns (specifically flocks) represent an aspect of the movement behavior of a group of moving entities, we can view patterns as a representation of collectives. Consider flocks, for instance, which can be described as a specific type of collective. It is for this reason that an ontology for collectives was chosen and hence, the set of collective features enumerated in [92] can be used to identify the semantic attributes that are suitable for interpreting flock patterns.

Out of existing ontologies related to collectives, we chose Wood and Galton's work [92] since its central focus is on collectives unlike other ontologies that represent very general domains, such as DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) [56], which describes ontological categories that underlie human common sense. Another example is BFO (Basic Formal Ontology) [13], which supports ontologies developed for scientific research. In contrast, other works such as [17] concentrate on restrictive types of collectives.

Due to the focus of [92] on collectives, it provides a more complete set of criteria for classification of collectives compared to others. For example, there is no way to model the role played by different members of a collective, or the reason for deeming a group as a collective with the other previously mentioned ontologies [17, 56, 13]. Some examples of collectives that are quite related to flocks and are described in [92] are protest march without leaders, protest march with leaders, and herd of cattle grazing in a field. Their description based on the criteria are as follows:

- A protest march without leaders - A collective with variable membership but cardinality > 1 . *Members are individuals which are not differentiated by role* and whose coherence is due to an internal collective purpose. The motion of the collective and its members are co-ordinated.
- A protest march with leaders - A collective with variable membership but cardinality > 1 . *Members are individuals which follow a hierarchical role model* and whose coherence is due to an internal collective purpose. The motion of the collective and its members are co-ordinated.
- A herd of cattle grazing in a field - A collective consisting of a constant set of individuals which are not differentiated by role and whose coherence is due to an external cause arising from an external purpose. The motion of the collective and its members are not co-ordinated.

In addition, our definition of a moving flock can be described using the criteria as follows: A collective with fixed membership but cardinality > 1 . Members are individuals which may be differentiated by role and whose coherence may be due to an internal or external collective purpose. The motion of the collective and its members are co-ordinated.

The Criteria for Semantic Attribute Selection We now describe the set of criteria in [92] in more details with respect to how each criterion can be used for semantic attribute selection. The general idea is to relate the candidate attributes to these criteria and disregard them if they do not match any of the criteria.

1. Membership - The identity of individual members and the cardinality of the collective are important features of the collective. Thus, attributes that uniquely identifies the members and those that describe the characteristics of the moving entities can be considered as candidate semantic attributes. Moreover, the size of the collective is also considered important. Candidate semantic attributes based on this criterion would include the members of the collective, the size of the collective and the characteristics of each member. Specific examples of individual characteristics in the context of people are his/her age, gender, occupation, health condition, and hobbies.
2. Location - A collective can be classified based on its location, the location of its members, and the relation between these two (i.e., movement of the members with respect to the collective). The location of the collective and the member can either be fixed or variable. Meanwhile, the movement of the members is either co-ordinated (i.e., moving in the same direction with relatively similar speed) or not. This demonstrates the importance of information about the collectives' location. Thus, geographic information can serve as candidate semantic attributes. Examples of such information are the interesting attractions visited by a collective, or the identifiable compartments of the attractions visited by the individuals. More specifically, examples of attractions are museums, hotels, and restaurants in the tourism domain. Meanwhile, examples of identifiable compartments in a museum are the different floors/levels of the museum or its individual rooms.
3. Coherence - A collective can be classified based on the source of the collective's coherence (i.e., the reason that unifies the entities as a collective). This source can come from the intentions of specific members or the collective. It can also originate from intentions of external agents or forces in the collective's environment. As a consequence, attributes that describe intentions or purpose of an individual, of the collective, or influencing factors in the environment can serve as candidate semantic attributes. Some examples of such attributes are the purpose of an individual in moving from one location to another, its activity during movement, the geographic characteristics of the collectives' spatial environment (ex: presence of obstacles or availability of pathways), and the characteristics of interacting entities with the collective or the members. The common properties of members, such as being teenagers or being accompanied by a dog, are also more specific examples of attributes describing source of coherence.
4. Roles - Members of a collective may be differentiated by roles. A collective can have hierarchical type of roles wherein some members play special roles that are structured in a hierarchy. Another possibility is having partitioned type of roles wherein there is a small number of differentiated roles played by many members. Lastly, roles can be individualistic wherein each member plays a specific role. Thus, attributes describing the roles of members in a collective and their relations with each other can be identified as candidate semantic attributes. Some examples of which are the occupation of an individual, or being a mother of another entity.
5. Depth - Members of a collective may be collectives by themselves. For example, individual trajectories in the DNP dataset may represent an individual, a couple, a

family, or other group types. Attributes that give this type of information can be selected as candidate semantic attributes.

As indicated by the use of the word *candidate* in the discussion, further attribute selection is necessary to identify the semantic attributes relevant to the required analysis task.

3.2.3 Two Levels of Semantic Annotation

We have also explicitly introduced two levels of semantic annotation, namely, the individual and the flock level. Due to the fact that a group of moving entities are involved in a flock pattern instance and the fact that the identity of these individual entities contribute to the existence of the flock, the individual trajectories involved in the flock instance should be semantically-enriched with their corresponding moving entity characteristics. Some examples of such characteristics are the age, the job description, and the intent of the moving individual.

In this step, we propose two options for annotating individual trajectories. First, all the trajectories can be annotated, especially in the case of dealing with a small and sparse dataset, so as to be able to infer statistically significant analysis results. Second, only the trajectories involved in the discovered patterns can be annotated. This provides the advantage of considering only a subset of the trajectories, resulting in shorter processing time during the annotation and analysis steps. Moreover, this allows the user to focus on relations that exist among the flocking entities.

Aside from annotating individual trajectories, the flock pattern themselves should also be semantically annotated. In fact, a flock pattern is an example of a collective, which has properties that only exist at an aggregate level. Examples of such properties are the spatial extent covered by the flocks as a whole, the start time and the end time of flocking, and the age range of the trajectories involved in discovered flocks.

An important point to notice here is that some flock level attributes are not directly available as individual level attributes are. Consider the age attribute at flock level, for instance. What is the value of the age attribute at the flock level? Can we simply use the mean in this case? To address this issue in the general case, we categorize the flock level attributes into three groups: (1) the parameters used by the flock discovery algorithm, (2) the flock descriptions generated by the flock discovery algorithm, (3) and the aggregated semantic properties of moving entities involved in the flock pattern. The first two groups of attributes can be directly obtained from the algorithm but the last group requires aggregation of individual level attributes before obtaining the semantic attributes at flock level. The aggregated attribute is obtained by first extracting all the possible values of an individual level attribute. An aggregated attribute is created for each possible value of the individual level attribute. For example, if we have an individual level attribute ‘loves_the_sun’ having two possible values ‘true’ and ‘false’, then there should also be two aggregated attributes, namely ‘loves_the_sun_true’ and ‘loves_the_sun_false’. The value of an aggregated attribute is then computed as the ratio between the number of moving individuals involved in the flock pattern satisfying the specific value of the individual attribute considered, and the total number of moving entities associated with the flock pattern. Continuing with the previous example, if there are 5 flock members in the currently considered flock with 3 members having the value ‘true’ and 2 members

having the value ‘false’ for ‘loves_the_sun’, ‘loves_the_sun_true’ for this flock will have the value 0.6 (i.e., computed from $3/5$) and ‘loves_the_sun_false’ will have the value 0.4 (i.e., computed from $2/5$).

Upon the completion of this second phase, the discovered flock patterns and the considered trajectories would be enriched with semantic data necessary for flock interpretation.

3.3 Pattern Analysis Phase

The last phase of the framework, which is the pattern analysis step, is concerned with understanding the occurrence and nature of the discovered patterns. In this phase, two data mining techniques are used for the analysis of the discovered patterns and their related attributes. These techniques are hierarchical clustering and classification with decision tree. For a concrete explanation of the application of these techniques, the remaining part of this section elaborates on how these are utilized in the interpretation of moving flock patterns.

3.3.1 Hierarchical Clustering for Pattern Analysis

For the hierarchical clustering step, correlation scores are first computed among the attributes and among the flock instances. These scores are then used to build the distance matrix needed to perform clustering. This analysis step produces dendograms that give an overview of the relations among the considered attributes and flock instances. The classification step, on the other hand, produces decision trees that focus on certain relations connecting the membership of entities to a specific flock with related semantic attributes. Though the decision trees do not cover all considered attributes nor all flock instances, unlike the dendograms, they provide more details as to how the discovered relations are connected.

Correlation Computation for Individual and Flock Attributes

For computing the correlation scores among individual and flock level attributes, we propose to use SUC (Symmetrical Uncertainty Coefficient) [39] and Pearson’s correlation coefficient [71]. We chose to use Pearson’s correlation coefficient since it is a standard measure for computing correlation scores. We propose the use of SUC as well since this allows us to check the consistency of correlation scores computed using different measures. We selected SUC since it considers the information entropy of attributes in the computation of their correlation scores. Moreover, it is also applicable when dealing with non-numeric attributes, unlike Pearson’s correlation coefficient, which is only applicable to numeric attributes.

Two variations of the Pearson’s correlation coefficient can be used as well. Taking the computed correlation score as is entails that only high positive correlations are considered as strong similarities among the attributes while high negative correlations are considered as strong dissimilarities. This is practical when there is a large number of attributes to analyze since the user can concentrate on the positive correlations. On the other hand, when the absolute value of the computed score is taken, both high positive and high negative correlations are considered as strong similarities. This is useful in scenarios wherein negative correlations are also considered important.

Given two attributes A_1 and A_2 , the correlation score between them using SUC is computed by applying Eq. 3.1.

$$SUC(A_1, A_2) = 2 * \frac{H(A_1) - H(A_1|A_2)}{(H(A_1) + H(A_2))} \quad (3.1)$$

where $H(X)$ refers to the entropy of the attribute X . The entropy is defined as the measure of uncertainty or randomness of the attribute X . If the attributes A_1 and A_2 are closely related, the conditional probability $H(A_1|A_2)$ found in the numerator will have a small value since knowing A_1 when A_2 is already previously known only gives few additional information. As a consequence, the numerator and hence the SUC as well will tend to have large values.

Meanwhile, the correlation score based on Pearson's correlation coefficient can be computed by applying Eq. 3.2.

$$\rho_{X,Y} = \frac{cov(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad (3.2)$$

where cov refers to covariance, E to expected value, μ to the mean, and σ to the standard deviation. X and Y are the distribution of the two attributes being compared. Pearson's correlation coefficient can be approximated based on a set of samples with size n by using Eq. 3.3.

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (3.3)$$

where X and Y are the distribution of sample values of the attributes being compared, while \bar{X} and \bar{Y} are the mean of the sample distributions.

This type of correlation analysis on individual attributes can reveal relations between specific flocks and particular features exhibited by their members. For example, individuals associated with a specific flock may belong to a particular age group while members of another flock may have the intent of shopping for clothes. In this case, the computed correlation between the first considered flock and the age group, and the other flock and the shopping intent should have large values.

Meanwhile, performing correlation analysis on the flock attributes allows the inference of associations among the flock properties themselves. For instance, flocks in which most members visit at least one museum may also have several members who are students.

Correlation Computation for Flock Instances

Computation of correlation scores for flock instances requires another type of measure since a pair of flocks has a different composition compared to a pair of attributes. In other words, each flock instance consists of a set of values for different attributes while each attribute is composed of a distribution of values.

The distance between a pair of flocks can be computed using two proposed techniques, either by basic distance computation or by taxonomy-based distance computation.

Basic Distance Computation Given a pair of flocks to be compared, we propose to directly compute their distance score by considering all corresponding attribute values and computing their difference. Obviously, for corresponding numeric attributes of the flock pair, the difference of their values can be computed in a straightforward manner and then normalized by dividing the result by the maximum difference between values of the considered attribute among every possible flock pairs. Analogously, when the attribute is nominal, a comparison test is performed in order to check whether the attributes values are equal or not, which is represented by a pair of fixed numeric values. Thus, each flock attribute produces a normalized distance score for every pair of flocks. Summing up the computed distance for each flock attribute makes up the distance between a flock pair.

However, this distance measure has some drawback since some attributes may be strongly-related with other attributes. As a result, the effect of such group of attributes on the distance score is amplified. To address this issue, a variation for computing the distance scores can be obtained by applying weights to each attribute depending on how strongly-related the attributes are, such that more strongly-related attributes are assigned lower weights. The motivation for this is to reduce the doubling effect that similar attributes have on the resulting distance. Instead of simply summing up the normalized distance for each attribute, the single normalized distances are first multiplied to their corresponding weight before summing them up.

Application of Weights We applied two types of weights: one is based on hierarchical clustering result that uses the symmetrical uncertainty coefficient as similarity measure while the other is based on the clustering result that uses the correlation coefficient measure. Using a distance matrix based on any similarity measure as parameter to the hierarchical clustering function in the R project [77] produces a merge and a height component. The merge component describes which pair of attributes or attribute sets (in the case that sub-clusters are combined) is clustered together at a time. These pairs are sorted according to the order that they were merged from the most similar to the least similar pairs. Meanwhile, the height component describes the corresponding distance value between the pair of attributes that were merged. Exploiting these components, the distance (found in the height component) between each pair of correlated attributes (found in the merge component) can be utilized as the weight of the attributes. In other words, the weight of an attribute is the distance between the attribute and the attribute most similar to it (i.e., having the shortest distance from it). Thus, attributes closely related to other attributes will be assigned lower weights. Moreover, the weight is normalized by using the maximum distance found in the height component as the denominator. A value of 1 is added to the numerator and the denominator. Adding 1 to the numerator ensures that the weight cannot be 0. Thus, totally removing the effect of attributes that are exactly similar with other attributes is avoided. On the other hand, adding 1 to the denominator ensures that the maximum weight computed is at most 1.

A Taxonomy-based Distance Computation A more sophisticated approach to computing flock similarity based on their properties is through the use of a taxonomy that describes the relation (usually *is-a* relation) among a subset of these properties. A problem with the approach described for computing correlation (or similarity) between flock instances is that matching of corresponding attributes and computing their difference is performed only among exactly corresponding attributes. This means that in the case that

there are correlated flock attributes, they are not compared despite of the correlation. This is addressed by building a taxonomy that shows which of the attributes are correlated. Aside from building the taxonomy, the names of the attributes had to be renamed in order to match those found in the taxonomy. After completing this preprocessing step, a straight-forward matching of values is performed by comparing exactly corresponding flock attributes and computing the difference between their values. Then, among the remaining attributes, the taxonomy is used to compute the similarity between each pair of flock attributes by considering their common ancestor that has the lowest position in the taxonomy compared to other possible ancestors for each pair. This ancestor is called the least common subsumer. We use Lin’s similarity measure [54], which is the ratio between the entropy of the least common subsumer and the sum of the entropy of the considered attribute pair, to compute the similarity score. The values for the pair of attributes having the highest similarity score are compared and the difference between them is computed. The matched part of the values is multiplied by the distance score (i.e., $1 - \textit{similarity_score}$) to account for the difference between the compared attributes. The same step is performed for other correlated pairs of attributes until the similarity scores among all remaining pairs fall below a specified threshold. Finally, the average of the differences computed from values of exactly matching attributes and of similar enough attributes is obtained. This average is the distance between the flock pair considered.

We provide an example of the application of the described approach. The aim is to compute the similarity score between *Flock 1* and *Flock 2*, which have five attributes as shown in Table 3.1. Two of these attributes, *Bird watching site* and *Sheepfold*, are very similar to each other while the rest are not correlated. *Flock 1* has 1 member interested in bird watching sites, 2 members interested in sheepfold areas, and 1 member interested in nature. *Flock 2*, on the other hand, has 2 members interested in bird watching sites, 1 in sheepfold areas, and 1 in prayer areas.

	<i>Bird watching site</i>	<i>Sheepfold</i>	<i>Bench</i>	<i>Nature</i>	<i>Prayer area</i>
<i>Flock 1</i>	1	2	0	1	0
<i>Flock 2</i>	2	1	0	0	1

Table 3.1: The given pair of flocks to be compared.

STEP 1: Straightforward Matching The first step involves performing straightforward matching among values of corresponding attributes. A member is removed from each flock for the *Bird watching site* attribute, and one member as well for the *Sheepfold* attribute. Thus, *Flock 1* is left only with 1 member for *Bird watching site* and *Flock 2* is left with 1 member for *Sheepfold*. Moreover, attributes having 0 values for both flocks, such as *Bench* in this example, are removed. The result of performing this sub-step is shown in Table 3.2.

	<i>Bird watching site</i>	<i>Sheepfold</i>	<i>Nature</i>	<i>Prayer area</i>
<i>Flock 1</i>	0	1	1	0
<i>Flock 2</i>	1	0	0	1

Table 3.2: Result of performing straightforward matching.

STEP 2: Diagonal Matching The next step is to perform diagonal matching on the

values of similar attributes. Using the taxonomy, which is no longer shown here, the similarity scores among every pair of the 4 remaining attributes are computed. Say that the following assumptions are true for the given example: the user-defined similarity threshold is set to 0.6, the similarity score between *Bird watching site* and *Sheepfold* is 0.9, and the remaining pairs of attributes have similarity scores lower than the threshold. Hence, only the values of *Bird watching site* and *Sheepfold* are matched. These attributes have 1 matching member in common and this value is multiplied to the distance score between the pair since the matching in this case is not exact as in the straightforward matching (i.e., $1 * (0.1) = 0.1$, where 1 is the number of matching values and 0.1 is the distance between the matched attributes). The result obtained after performing diagonal matching is shown in Table 3.3.

	<i>Bird watching site</i>	<i>Sheepfold</i>	<i>Nature</i>	<i>Prayer area</i>
<i>Flock 1</i>	0	0	1	0
<i>Flock 2</i>	0	0	0	1

Table 3.3: The result after performing diagonal matching.

STEP 3: Averaging the Computed Distances In the case that there are other pairs of attributes having similarity scores higher than the 0.6 threshold, the same step is performed for every such pair. Since there is no longer such pairs of attributes, the diagonal matching phase is finished and we compute the straightforward difference among the remaining attributes with non-0 values for at least one of the flocks. The difference for the remaining attributes, *Nature* and *Prayer area*, is 1 each. We sum this up with the distances computed during the diagonal matching phase and compute the average based on the initial number of attributes considered, which is 5 in this example. The resulting value 0.42, which is obtained from $\frac{1+1+0.1}{5}$, is the distance score. We simply subtract this value from 1 to obtain a similarity score of 0.58. Obtaining this result is illustrated in Table 3.4.

	<i>Nature</i>	<i>Prayer area</i>	
<i>Flock 1</i>	1	0	
<i>Flock 2</i>	0	1	
	dist=1	dist=1	dist=0.1 (from diagonal matching)

Table 3.4: Obtaining the final similarity score between the given pair of flocks.

Computing the correlation among different flock patterns allows the analyst to identify similar flock types and focus on certain types, such as flocks whose members belong to a specific age group. Combined with the analysis of individual and flock attributes, it is possible to pinpoint specific attributes that make flocks similar to other flocks.

Hierarchical Clustering The clustering step is aimed at finding groups of attributes and groups of flock instances that are highly correlated. Hierarchical clustering, in particular, groups together entities in a progressive way such that the most highly correlated entities are first grouped together and this is applied recursively until the entire set of entities is grouped into one cluster. The clustering algorithm requires a distance matrix, which summarizes the dissimilarity scores among the items, as parameter. The distance matrix

for individual and flock level attributes can be easily computed from the correlation scores obtained in the correlation computation step for individual and flock attributes while the distance matrix for flock instances can be easily extracted from the distance scores computed in the correlation computation step for flock instances.

Hierarchical clustering is performed for the set of individual attributes, the set of flock attributes, and the set of discovered flock patterns. The output of the clustering step may be useful when the analyst needs to focus on certain groups of flock patterns that are of interest. This is remarkably useful when there are a large number of flocks discovered. Furthermore, clustering may reveal interesting relationships, which may not be obvious to a domain expert, and they may support the analyst in uncovering possible reasons for the occurrence of the mined flock instances.

The correlation computation step combined with the hierarchical clustering step produces dendrograms that give an overview of the relations among the set of individual attributes and among the set of flock attributes. The obtained results are still quite limited in terms of aiding the user in understanding the nature and occurrences of the flocks. Although clustering is able to pinpoint which attributes are correlated, it is not able to pinpoint how or why these attributes are correlated.

3.3.2 Classification for Pattern

We have extended the flock analysis step further by performing individual attribute classification based on flock membership attributes and flock attribute classification based on some interesting flock properties.

We propose the use of a decision tree based classification algorithm in order to study the relations among each flock membership attribute and the other individual attributes since decision trees are simple and intuitive.

For the individual attribute classification, we set the class attribute to an individual flock membership attribute. For example, we set the class attribute to be *flock0*, which refers to whether or not an individual belongs to *flock0*. We have specifically selected a cost-sensitive version of the J48 classification algorithm, whose implementation is accessible in WEKA [40], to generate the decision tree connecting individual attributes that contribute to the membership of an individual to a specific flock.

As for the flock attribute classification, we set the class attribute to an interesting flock property. For instance, the main activity of the flock can be of interest to park managers and hence, it would be useful to understand how this is related to other flock attributes. Specifically, the class attribute can be set to *main_activity_1*, which refers to the percentage of flock members walking.

The use of a cost-sensitive algorithm is appropriate when the dataset considered consists of entries biased to a specific value. Putting more weights to the less occurring value eases the bias present in the dataset.

J48 is WEKA's Java implementation of the C4.5 algorithm, which in turn is a known standard classification algorithm. It was developed by Ross Quinlan as an extension of his earlier ID3 algorithm. The algorithm builds decision trees by considering the attribute that most effectively splits the training dataset into subsets having the most homogenous values for the target attribute. This is determined by computing each attribute's normalized information gain and splitting the dataset using the attribute with the highest normalized information gain. This step is applied recursively to the subsets obtained at each step.

3.4 Summary of Discussion and Conclusions

We have presented the proposed pattern interpretation framework in this chapter and have described how it deals with the data mining and pattern interpretation phases of KDD for handling movement data. The framework is able to encompass two main phases of KDD by providing phases that handle both extraction and interpretation of patterns. The pattern discovery phase deals with pattern extraction while a combination of the semantic annotation and the pattern analysis phases allow interpretation of patterns.

Aside from providing a conceptual framework, we have also discussed how the framework can be instantiated and implemented for moving flock patterns.

Pattern Discovery Phase In order to realize the pattern discovery phase, a clear definition of the pattern and a corresponding algorithm as well are necessary. Since moving flock is a new concept, its formal definition was formulated and we have initially introduced it in [89]. Moreover, we have also developed and implemented an algorithm, which was also initially presented in the same work. The algorithm consists of four steps (synchronization, spatial neighbor computation, membership persistence analysis, and pruning), which deal with synchronized sampling of points, checking for spatial and temporal coherence, and filtering of redundant and stationary flocks.

Furthermore, other issues related to the selection of the algorithm's input parameters and its validation were also managed. A technique used for DBSCAN was adapted to handle the selection of the *radius* parameter while general guidelines were provided for the rest of the parameters. In validating the algorithm, we have observed visualizations of the flock trajectories to assess if the plots are reasonable. Moreover, we have introduced a more sophisticated technique, which is based on the null hypothesis principle. This technique involves randomization of the input dataset, running the algorithm on the randomized dataset, and comparing the obtained results with those obtained from the original dataset. A large difference among the compared results would validate the fact that the patterns extracted by the algorithm are inherent in the dataset, and not by mere chance.

Semantic Annotation Phase As for the semantic annotation phase of the framework, issues related to the possible sources of semantic information, the selection of semantic attributes to be used for annotation, and the appropriate level of annotation were also dealt with. Possible sources of semantic information are questionnaires/surveys, thematic and topographic maps, information on POIs, census data, and many others.

Aside from having many possible sources of semantic information, a single source itself is likely to contain many attributes as well. As a consequence, selecting appropriate attributes for the annotation phase becomes a challenge. To address this issue, we proposed a guideline based on Wood and Galton's criteria for classification of collectives [92]. It is important to note, however, that further application of other attribute selection techniques must be applied to finalize the set of selected semantic attributes.

Once the semantic attributes have been selected, the next challenge involves enrichment of movement data and patterns with these attributes. This enrichment must be performed at an appropriate level and for this purpose, we propose a combination of two levels of annotation, namely individual and pattern level. While individual annotation can be easily performed based on attributes already present in the movement data, pattern annotation

is not as straight-forward. We proposed the use of an aggregation technique for this level of annotation.

Pattern Analysis Phase For the final phase of the framework, we propose the use of both hierarchical clustering and decision tree classification algorithms for deducing the relations among the individual attributes, among the pattern attributes, and among the patterns themselves. The result obtained from the application of the two algorithms to these attributes and patterns can provide support for understanding the relations between members and the patterns (or flocks) they belong to, the relations among members of the same pattern, and the relations among the discovered patterns. This, in turn, can lead to understanding the interactions involved among the moving entities, and their interactions with their environment.

The next chapter provides a detailed discussion of the application of the framework to real-world datasets for the purpose of moving flock interpretation in the context of pedestrian movement.

Chapter 4

Experiments

This chapter provides a summary of the datasets used to evaluate the feasibility and effectiveness of the proposed framework. It also contains a discussion of the performed experiments and the obtained results.

4.1 Datasets

Three datasets, which were briefly described earlier in Chapter 1, were used in order to test the implemented moving flock discovery algorithm and the effectiveness of the proposed framework for flock interpretation. These include the DNP, the Fontainebleau, and the Delft datasets. All three of which are pedestrian datasets. While the entire framework was tested on the DNP and the Delft datasets, it was not fully tested on Fontainebleau since we did not have access to its semantic attributes, which are needed for the annotation and the interpretation phases.

Table 4.1 provides an overview of the datasets. DNP and Fontainebleau describe movement in recreational parks while Delft describes movement in a city setting. The summary shows how movements in DNP are faster compared to those in Delft and thus, average distances between spatial points in DNP are also larger. The Fontainebleau dataset, though, contains strange values for the sampling rate, the average speed, and the average distance due to the large amount of noise present in the raw dataset.

It is also worth noting that some pedestrians may have more than one trajectory in the datasets. This is due to the preprocessing step performed on the data using M-Atlas. The intuition behind this preprocessing step is that a person is most likely starting a new trip, or some error occurred in collecting the observation points when any of the thresholds (as described in Section `subsec:movingFlock`) are exceeded. For example, a point is considered as a noise when it is too far from its preceding point, implying that such movement is unfeasible for pedestrians. Furthermore, long time gaps occurring within a trajectory likely represents stops in a workplace or other interesting area, and can serve as markers for different types of trips, such as a trip to work or to the supermarket.

DNP Semantic Attribute Source The DNP dataset is an interesting case study despite of its small size since it contains attributes derived from visitors' responses to the conducted survey, which contains 23 questions from which 73 visitor attributes were derived. Whether the visitor is on holiday, the frequency of visit, the number of accom-

	<i>DNP</i>	<i>Fontainebleau</i>	<i>Delft</i>
<i>Description</i>	a Dutch recreational park containing networks of short and long trails, dry and wet heath lands, pine and deciduous forest and complex of juniper shrubs. It also includes sheep farms, some bird-watching lookouts, staffed and unstaffed information centres, a tea house and cultural spots (ex: a historical house and a radio telescope)	a French recreational forest park that has a massive wooded area. Its wild landscape attracts a considerable number of hikers, rock-climbing or mountain-biking enthusiasts, horse riders, cyclists and Sunday walkers [30]. Its routes are usually used for walking while its forest contains wild plants and trees, and a population of birds, butterflies and mammals.	a Dutch city known for its town center and its canals. It contains different types of attractions such as churches, museums, factories, windmills, gardens, libraries, markets, restaurants, bars, cafes, and shopping areas. Some examples of its historical buildings are Oude Kerk, Nieuwe Kerk, Prinsenhof, its city hall, Oostpoort, Gemeenschapshuis, and Waag [23]. It also houses the Delft University of Technology.
<i>Trajectories</i>	370 trajectories (141,826 sampling points) of 372 visitors (around 1.29% of actual visitors)	207 trajectories (22,748 sample points) of 23 visitors (around 0.015% of actual visitors)	303 trajectories (467,454 sample points) of 285 pedestrians (around 2.6% of actual visitors)
<i>Days</i>	May 18, 25, 28, August 6, 9, 17, 19, 2006 (7 days; semi-synthetic version was collapsed to 1 day)	April 25, May 11, September 24-26, 2004 (5 days)	November 18-21, 2009 (4 days)
<i>Dataset Sampling Rate</i>	Random; average = 16.52s	Random; average = 143.22s	Around 2s; average = 3.7s
<i>Synchronization Rate</i>	5 mins.	1 min.	1 min.
<i>Trajectories per day</i>	May 18 2006 - 38 trajs.; May 25 2006 - 56 trajs.; May 28 2006 - 88 trajs.; August 6 2006 - 81 trajs.; August 9 2006 - 37 trajs.; August 17 2006 - 40 trajs.; August 19 2006 - 30 trajs.	Apr 25 2004 - 109 trajs.; May 11 2004 - 1 traj.; Sept 24 2004 - 2 trajs.; Sept 25 2004 - 423 trajs.; Sept 26 2004 - 1177 trajs.	Nov 18 2009 - 58 trajs.; Nov 19 2009 - 109 trajs.; Nov 20 2009 - 101 trajs.; Nov 21 2009 - 35 trajs.
<i>Average Speed</i>	1.37 m/s	60.2m/s	0.87m/s
<i>Average Distance</i>	14.58m	1083.388m	1.9m
<i>Selected Semantic Attributes</i>	on_holiday, freq_visit, since_when, adult_num, children_num, visitor_type, main_activity, attraction_visited, picnic_areas, mound, info_centre, bird_watching_site, prayer_areas, juniper_berries, fens, sheepfold_areas, sightseeing_areas, radio_telescope, david_lakes, orienting, tea_houses, route, parking_access, sheepfold_proximity, attraction_proximity, route_start, coincidence, catering, beautiful, quiet, seat, lunch, nr_information, white_route, whitelheederzand, redspier, local_living, age_category	N/A	purpose, shopping, first_visit, frequency, postcodeb, originb, gender, age, group_, occup, household, wth_sunny, wth_cloudy, wth_rainy, wth_rain, wth_windy

Table 4.1: Description of datasets used for the experiments.

panying children, adults and dogs, and the main attractions visited are some examples of such attributes. The following is a subset of the survey questions from which these attributes were derived:

1. Are you in Dwingelderveld for vacation?
Yes No
2. How often you come to DNP? (1 tick)
 - daily
 - weekly
 - monthly
 - 2-4 times per year
 - 1 time per year
 - Today is the first time
 - Others (please specify): _____
3. Who are you with today in the park?
I'm with ____ adults (including yourself) and ____ children
With ____ dog (s)
4. Which of these places /services have you visited today?
(several answers possible)
 - Picnic areas
 - Mound
 - Bird watching site
 - Prayer areas
 - Others (please specify): _____

4.2 Moving Flock Discovery

The moving flock algorithm was implemented using Java on a PC with Windows XP OS, Intel Pentium 4 and 2 GB of main memory. This section is further divided into subsections that covers a discussion of the moving flock results, the plotted line graphs for the selection of the *radius* parameter, the effect of using different radius values, the effect of varying the order in the input file, and the validation of the algorithm.

4.2.1 Moving Flock Results

This subsection provides a discussion of the obtained moving flocks given the three input dataset: DNP, Fontainebleau, and Delft. We have used the following parameter values for defining the flocking behavior in the experiments: *min_points* to 3 members, *min_time_slices* to 3 time instances, *synchronization_rate* to either 1-minute or 5-minute sampling rate, and the *radius* to different values in meters.

DNP Flock Results

Table 4.2 shows the moving flocks obtained from the DNP dataset using three radius values (i.e. $radius=100m$, $150m$, and $200m$) and a 5-minute synchronization rate. The results shows that the larger the radius values, the greater the prospect of discovering a larger number of moving flocks. However, this is not always true, since our algorithm filters out stationary flocks based on the $radius$ parameter and as the radius becomes larger, flocks with short extent are filtered out.

<i>Radius</i>	<i>Start Time</i>	<i>End Time</i>	<i>Flock Extent</i>	<i>Flock Members</i>
100	Sun May 28 12:45:00 2006	Sun May 28 13:00:00 2006	203.9375	139; 141; 140
150	Sun May 28 09:40:00 2006	Sun May 28 10:00:00 2006	628.5	52; 100; 46
	Sun May 28 12:45:00 2006	Sun May 28 13:00:00 2006	203.9375	139; 141; 140
200	Sun May 28 09:40:00 2006	Sun May 28 10:05:00 2006	796.5	52; 100; 46
	Sun May 28 13:40:00 2006	Sun May 28 13:55:00 2006	456	115; 118; 114
	Sun May 28 12:45:00 2006	Sun May 28 13:00:00 2006	203.9375	139; 141; 140

Table 4.2: Discovered moving flock patterns in DNP.

Figure 4.1 illustrates one of the discovered moving flocks containing the three members 139, 141 and 140. This flock was found using all the three specified radius values: $100m$, $150m$, and $200m$.

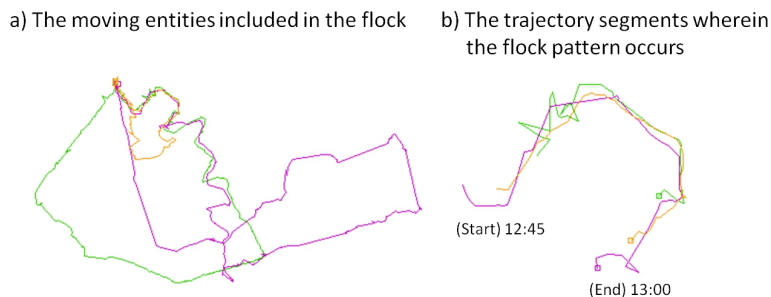


Figure 4.1: (a) The entire trajectories of moving flock members 139, 141 and 140 in the DNP dataset. (b) The trajectory segments belonging to the moving flock whose members include 139, 141 and 140 in the DNP dataset.

Due to the sparseness of the original DNP dataset, we have also generated a semi-synthetic version of the dataset, which is basically the same as the original DNP dataset but with the dates collapsed to one day. This was done for the purpose of properly testing the algorithm on a denser dataset.

Running the algorithm on this dataset produced a larger number of moving flocks. 11 moving flocks were found when the radius was set to $150m$, and 34 moving flocks were found when the radius was set to $200m$.

The top three moving flock patterns ranked by extent for radius $150m$ and $200m$ are found in Table 4.3. Note that the top two flocks are the same for both radii. Meanwhile, the 3rd top flock for radius 150 is also found when the radius is set to $200m$ but with a minor difference in time and extent. Same is true for the 3rd flock for radius 200 .

The bottom three flocks, on the other hand, for the same radii are found in Table 4.4. Note that the 3rd to the last flock of radius 150 is similar to the last flock of radius 200 . The 2nd to the last flock is also found when the radius is set to 200 but with minor differences due to the larger radius used. The last flock of radius 150 is similar to some flocks but having only 3 flock members when the radius is set to 200 . The 3rd to the last

Radius	Start Time	End Time	Flock Extent	Flock Members
150	Thu Dec 30 12:15:00 1999	Thu Dec 30 12:30:00 1999	991.9375	96; 288; 15
	Thu Dec 30 09:40:00 1999	Thu Dec 30 09:55:00 1999	870.5	228; 287; 104
	Thu Dec 30 11:55:00 1999	Thu Dec 30 12:05:00 1999	692.4375	118; 249; 346
200	Thu Dec 30 12:15:00 1999	Thu Dec 30 12:30:00 1999	991.9375	96; 288; 15
	Thu Dec 30 09:40:00 1999	Thu Dec 30 09:55:00 1999	870.5	228; 287; 104
	Thu Dec 30 09:40:00 1999	Thu Dec 30 10:05:00 1999	796.5	52; 100; 46

Table 4.3: Top three moving flock patterns in the semi-synthetic version of the DNP Dataset.

flock of radius 200 is also found when the radius is set to 150 but the 2nd to the last flock is not.

Radius	Start Time	End Time	Flock Extent	Flock Members
150	Thu Dec 30 12:45:00 1999	Thu Dec 30 13:00:00 1999	203.9375	139; 141; 140
	Thu Dec 30 12:40:00 1999	Thu Dec 30 12:50:00 1999	178.5	139; 365; 140
	Thu Dec 30 12:50:00 1999	Thu Dec 30 13:00:00 1999	150.125	140; 139; 142; 129
200	Thu Dec 30 11:20:00 1999	Thu Dec 30 11:30:00 1999	209	158; 203; 78
	Thu Dec 30 12:45:00 1999	Thu Dec 30 12:55:00 1999	207.625	127; 365; 187
	Thu Dec 30 12:45:00 1999	Thu Dec 30 13:00:00 1999	203.9375	139; 141; 140

Table 4.4: Bottom three moving flock patterns in the semi-synthetic version of the DNP Dataset.

Figure 4.2 shows the base trajectories for each of the discovered moving flocks with the radius set to 150m in the semi-synthetic dataset and visualizing them on top of the set of all trajectories (Figure 4.2a) and on a Google map (Figure 4.2b) as background. The balloons in the Google map indicate the ending point of the flock trajectories. Based on the figures, most of the flocking occurred on the west side of the park.

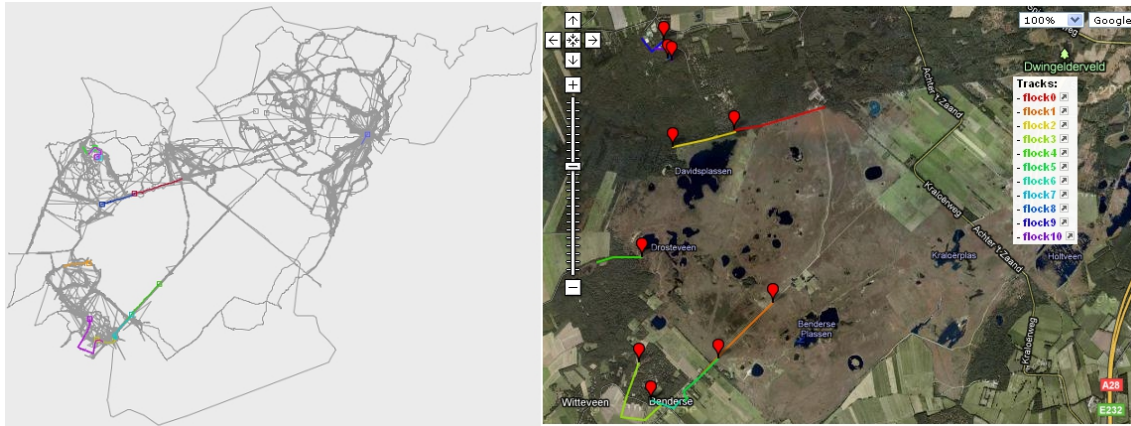


Figure 4.2: The base trajectories of the moving flock patterns found in the semi-synthetic version of the DNP dataset when the radius is set to 150m using (a) the whole trajectory dataset and (b) a Google map as background.

Finally, a large number of stationary flock patterns were filtered out by our algorithm as shown in Figure 4.3 where the number of moving flocks and the number of stationary flocks increases with respect to the user-specified radius (until around 900m) for this dataset.

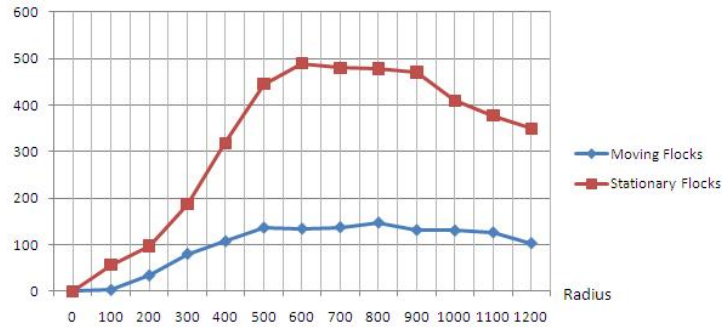


Figure 4.3: The number of moving flocks versus the number of stationary flocks in the semi-synthetic version of the DNP dataset.

Fontainebleau Flock Results

This part discusses the results obtained from the Fontainebleau dataset using the moving flock discovery algorithm. Table 4.5 shows the different time durations and spatial extents of the moving flocks obtained from the Fontainebleau dataset when the radius was set to 150m and 200m. The 1st flock of radius 150 and the 2nd flock of radius 200 are quite similar, having 2 members in common and having overlapping flock duration. The 2nd flock of radius 150 and the 1st flock of radius of 200 are more similar, having similar members but having a shorter time duration compared to that of radius 200. Plots of the moving flocks extracted when the radius was set to 150m are depicted in Figure 4.4.

Radius	Start Time	End Time	Flock Extent	Flock Members
150	Sun Apr 25 13:38:00 2004	Sun Apr 25 13:48:00 2004	159.34375	9; 11; 8
	Sun Sep 26 11:41:00 2004	Sun Sep 26 12:04:00 2004	150.28125	19; 27; 18
200	Sun Sep 26 11:38:00 2004	Sun Sep 26 12:18:00 2004	264.71875	19; 27; 18
	Sun Apr 25 13:34:00 2004	Sun Apr 25 13:41:00 2004	264.5	9; 10; 8

Table 4.5: Discovered moving flock patterns in the National Fontainebleau Forest Park.



Figure 4.4: Flock 0 and Flock 1 discovered from the Fontainebleau dataset using a radius of 150m.

An overall view of the discovered moving flocks when the radius is equal to 150m is found in Figure 4.5, which shows the base trajectories of the flocks, with the set of all trajectories (Figure 4.5a) and with Google map (Figure 4.5b) as background.

The number of moving flocks and the number of stationary flocks with respect to the specified radius is plotted in Figure 4.6. Once again, this demonstrates the large number of stationary flocks pruned out by our algorithm.

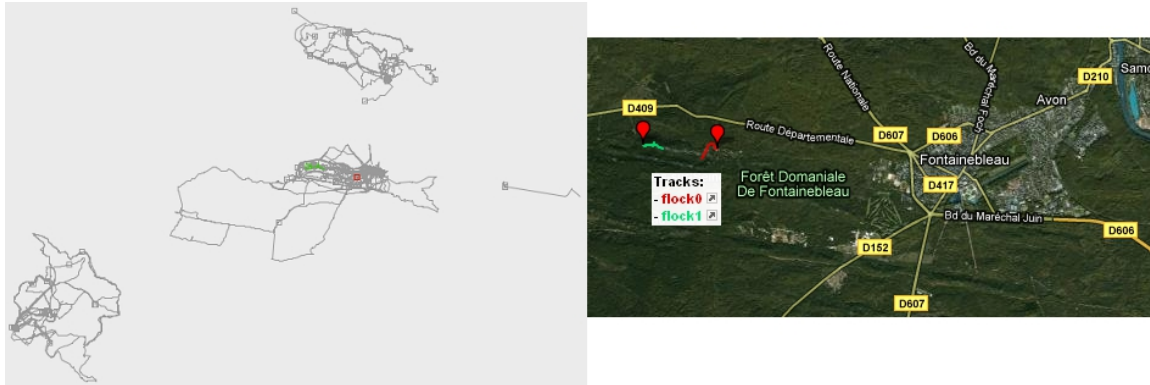


Figure 4.5: The base trajectories of the moving flock patterns found in the Fontainebleau dataset when the radius is set to 150m using (a) the whole trajectory dataset and (b) a Google map as background.

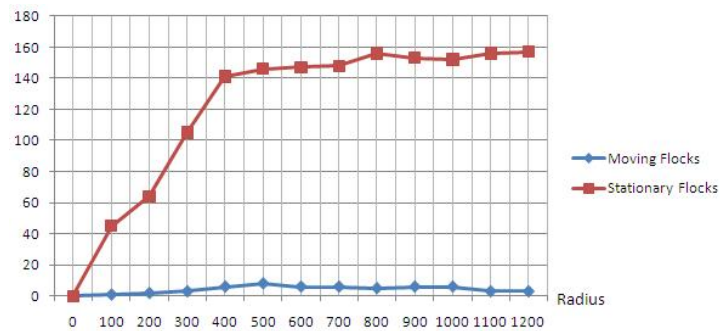


Figure 4.6: The number of moving flocks versus the number of stationary flocks in the Fontainebleau dataset.

Delft Flock Results

This part provides a description of the patterns obtained from the Delft dataset using the moving flock algorithm. Table 4.6 shows the different time durations and spatial extents of the moving flocks obtained from the Delft dataset when the synchronization rate was set to 1 minute and the radius to 40m and 50m. There were 3 flocks found when the radius was set to 40m and 10 flocks when the radius was 50m. All 3 flocks for radius 40m are also found when the radius is 50m as shown in the table. The first 2 flocks discovered using both 40m and 50m are exactly the same, while their third flock varies by 1 minute in the flocking duration and by few meters in the extent.

Figure 4.7 provides an overview of where the moving flocks occurred by presenting the base trajectories of each moving flock against the set of all trajectories (Figure 4.7a) and Google map (Figure 4.7b). The radius used to find these flocks is 50m. Meanwhile, Figure 4.8 provides a plot of *flock0* and *flock9* found in the Delft dataset using a 50m radius and a 60s synchronization rate.

As with DNP and Fontainebleau, a large number of stationary flock patterns were also filtered out by the moving flock algorithm in Delft as shown in Figure 4.9.

Radius	Start Time	End Time	Flock Extent	Flock Members
40	Thu Nov 19 11:12:00 2009	Thu Nov 19 11:14:00 2009	67.5	222; 713; 206
	Fri Nov 20 12:05:00 2009	Fri Nov 20 12:07:00 2009	65.5	315; 846; 303
	Thu Nov 19 10:17:00 2009	Thu Nov 19 10:20:00 2009	55	709; 712; 708
50	Fri Nov 20 10:33:00 2009	Fri Nov 20 10:35:00 2009	83	818; 819; 815
	Thu Nov 19 10:45:00 2009	Thu Nov 19 10:50:00 2009	70.5	212; 708; 203
	Thu Nov 19 11:12:00 2009	Thu Nov 19 11:14:00 2009	67.5	222; 713; 206
	Fri Nov 20 12:05:00 2009	Fri Nov 20 12:07:00 2009	65.5	315; 846; 303
	Thu Nov 19 10:16:00 2009	Thu Nov 19 10:20:00 2009	63.5625	709; 712; 708
	Thu Nov 19 11:19:00 2009	Thu Nov 19 11:27:00 2009	63.25	223; 721; 206
	Sat Nov 21 10:38:00 2009	Sat Nov 21 10:40:00 2009	61	408; 908; 901
	Thu Nov 19 10:44:00 2009	Thu Nov 19 10:48:00 2009	60.625	203; 715; 223
	Thu Nov 19 10:50:00 2009	Thu Nov 19 11:00:00 2009	55.5	223; 708; 221
	Fri Nov 20 13:45:00 2009	Fri Nov 20 13:50:00 2009	50.75	843; 862; 807

Table 4.6: Discovered moving flock patterns in the Delft Dataset when radius is set to 40m and 50m.

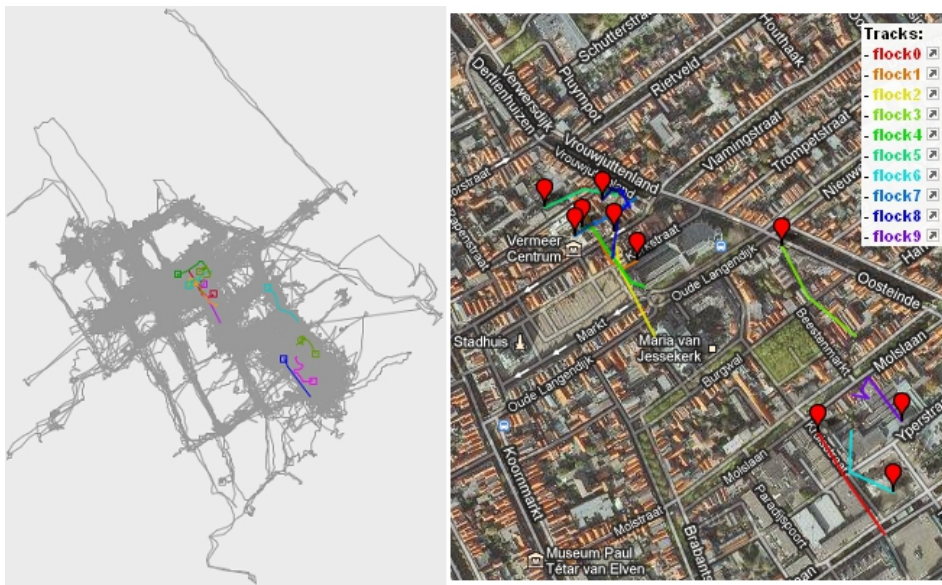


Figure 4.7: The base trajectories of the moving flock patterns found in the Delft dataset when the radius is set to 50m using (a) the whole trajectory dataset and (b) a Google map as background.

4.2.2 Selection of the *Radius* Parameter

For the purpose of guiding the user in setting the *radius* parameter of the moving flock algorithm, we have plotted the line graph that shows the distance of the objects from their k -th nearest neighbor for each dataset, where $k = 1 \dots 10$. The input required for building these graphs is the synchronized version of the datasets, containing (id, x, y, t) sampled points. The semi-synthetic version of the DNP dataset was synchronized to 5 minutes, while the Fontainebleau and the Delft datasets were synchronized to 1 minute. The choice of these synchronization values were based on the visualization of the synchronized points. While using 5 minutes for the DNP dataset already gave good plots that are quite close to the original trajectories, using the same value for the other datasets still introduced a significant amount of noise in the trajectories, causing the plots to vary quite significantly in certain parts compared to the original. After a few number of trial and error, we found 1 minute to be a suitable value for the Fontainebleau and Delft datasets.

The plots obtained for the DNP dataset using k values ranging from 1 to 10 are found in

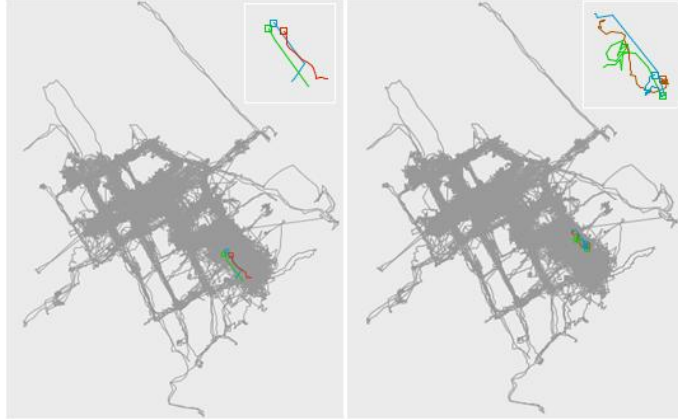


Figure 4.8: The highest-ranking (left) and the lowest-ranking (right) moving flocks discovered in the Delft dataset using 50m as the radius and 60s as the synchronization rate.

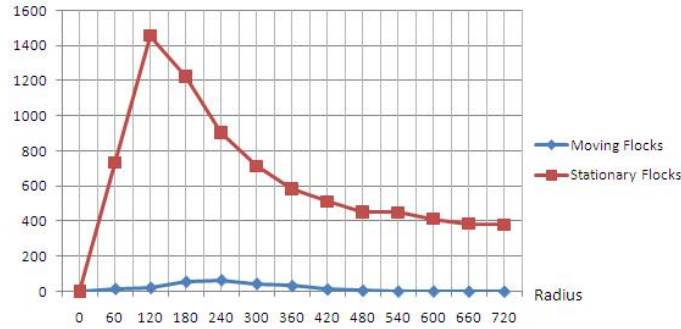


Figure 4.9: The number of moving flocks versus the number of stationary flocks in the Delft dataset.

Figure 4.10. When $k = 3$, The knee of the curve is at around 300m, suggesting that values close to 300m are good radius values. For applying the flock interpretation framework on this dataset, we chose to work with a radius value of 150m being guided by this plot and by the fact that a spatial closeness of 300m is quite large in the context of pedestrians.

Figure 4.11 provides the plots for the Fontainebleau dataset for varying k 's. For $k = 3$, the knee of the curve is approximately at 500m. Again, since 500m is too large to define spatial closeness among pedestrians, we chose 150m and 200m in testing the moving flock discovery on this dataset.

Lastly, the plots for the Delft dataset are shown in Figure 4.12. Considering $k = 3$ and comparing with the previous plots, the knee of the curve occurs at a low value of around 30m. Being guided by this, it is not reasonable to use 150m or 200m for this dataset since these values are too large compared to the suggested value. In the experiments, we chose to work with a 50m radius in order to take into account the GPS uncertainties present in the data.

This set of experiments shows that plotting the k -th distance is an effective technique for guiding the user in selecting the *radius* parameter. The technique does not give the exact radius value since it depends on other factors such as the nature of the moving entities, the property of the area in which movement was made, the uncertainties present

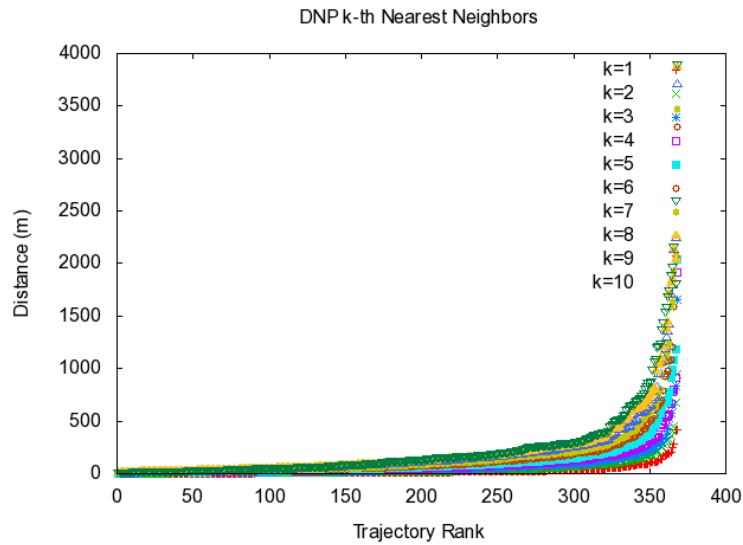


Figure 4.10: Plot of k -th Distances for the Semi-synthetic DNP Dataset.

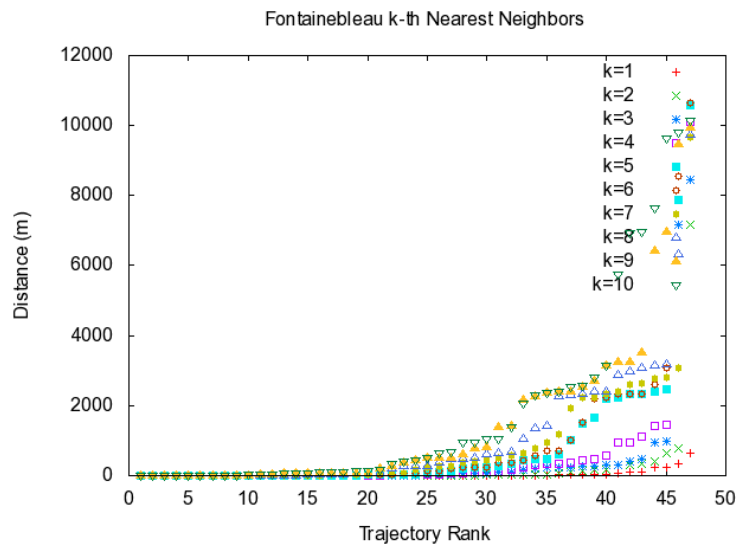


Figure 4.11: Plot of k -th Distances for the Fontainebleau Dataset.

in the data due to limitations of existing location technologies, and others. However, it is still useful in deciding if a radius value is too small or too large with respect to the dataset distribution. It has also been observed that plotting the nearest neighbors for varying values of k produced line graphs that are quite similar in shape but with larger distance values for larger k 's.

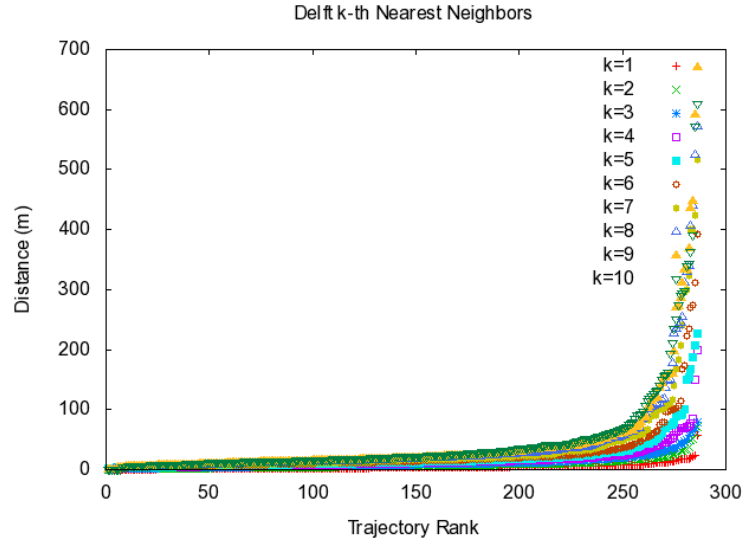


Figure 4.12: Plot of k -th Distances for the Delft Dataset.

4.2.3 Effect of Varying the Radius Value

In order to assess the effect of using different radius values on the moving flock algorithm, we ran it several times on a subset of the semi-synthetic DNP dataset focusing on trajectories occurring during lunchtime from 12:00 to 13:00. We chose to work on this smaller set of results for the ease of comparing flock results. The same set of parameters except for the *radius* were used for the different runs and are as follows: *min_points*=3, *min_time_slices*=3, *synchronization_rate*=300s, and *radius*=50m, 80m, 100m, 130m, 150m, 180m, 200m. Table 4.7 summarizes the flock results obtained using these parameters. It shows how the number of flocks increases as the radius value is increased. At a certain point though, if the radius becomes too large, the number of flocks should decrease due to the filtering of stationary flocks by the radius size.

Comparing the flocks obtained from each radius and the next higher radius, the table shows that the flocks obtained using a smaller radius is also obtained using the larger radius in most cases. The description of the obtained flocks may vary in time and thus, in extent as well. The composition of the flock may also vary in some cases, though it is not exhibited in these results. There is only 1 instance wherein a flock obtained using a certain radius is not found using the next higher radius in the table. This particular instance is the last flock of radius 150m with members 139, 140, 129, which is not found using a radius of 180m. The most probable reason for this is that though these visitors may flock together, the flock extent is lower compared to the 180m radius. However, this un-discovered flock (i.e., discovered using 150m but not with 180m) is overlapping with another discovered flock, which is the last flock (using 180m).

Aside from affecting the structure of discovered flocks, it is also important to note that the use of varying radius corresponds to finding flocks that varies in semantics as well. For instance, it is more appropriate to use a smaller radius value when the user is interested in finding flocks along the streets as opposed to flocks occurring in a block or in a city center. Thus, the appropriate choice of radius depends on the scale that the user is interested in.

Radius	Start Time	End Time	Flock Extent	Flock Members
50	No Flock			
80	Thu Dec 30 12:20:00 1999	Thu Dec 30 12:30:00 1999	655.9375	15; 288; 96
100	Thu Dec 30 12:20:00 1999	Thu Dec 30 12:30:00 1999	655.9375	15; 288; 96
	Thu Dec 30 12:45:00 1999	Thu Dec 30 13:00:00 1999	203.9375	139; 141; 140
130	Thu Dec 30 12:20:00 1999	Thu Dec 30 12:30:00 1999	655.9375	15; 288; 96
	Thu Dec 30 12:45:00 1999	Thu Dec 30 12:55:00 1999	203.9375	140; 141; 139
	Thu Dec 30 12:50:00 1999	Thu Dec 30 13:00:00 1999	150.125	129; 140; 139
150	Thu Dec 30 12:15:00 1999	Thu Dec 30 12:30:00 1999	991.9375	96; 288; 15
	Thu Dec 30 12:45:00 1999	Thu Dec 30 13:00:00 1999	203.9375	139; 140; 141
	Thu Dec 30 12:50:00 1999	Thu Dec 30 13:00:00 1999	150.125	139; 140; 129
180	Thu Dec 30 12:15:00 1999	Thu Dec 30 12:30:00 1999	991.9375	96; 288; 15
	Thu Dec 30 12:30:00 1999	Thu Dec 30 12:40:00 1999	306.8125	134; 288; 35
	Thu Dec 30 12:45:00 1999	Thu Dec 30 12:55:00 1999	258.6875	140; 141; 365
	Thu Dec 30 12:45:00 1999	Thu Dec 30 13:00:00 1999	203.9375	140; 141; 139
200	Thu Dec 30 12:15:00 1999	Thu Dec 30 12:30:00 1999	991.9375	96; 288; 15
	Thu Dec 30 12:20:00 1999	Thu Dec 30 12:30:00 1999	429.25	341; 346; 280
	Thu Dec 30 12:25:00 1999	Thu Dec 30 12:40:00 1999	410.5625	118; 249; 227
	Thu Dec 30 12:40:00 1999	Thu Dec 30 12:55:00 1999	328.625	140; 365; 139
	Thu Dec 30 12:30:00 1999	Thu Dec 30 12:40:00 1999	306.8125	134; 288; 35
	Thu Dec 30 12:45:00 1999	Thu Dec 30 12:55:00 1999	207.625	127; 365; 187
	Thu Dec 30 12:45:00 1999	Thu Dec 30 13:00:00 1999	203.9375	140; 141; 139

Table 4.7: Discovered moving flock patterns in a subset of the semi-synthetic DNP Dataset when the radius is set to varying values.

4.2.4 Effect of Ordering of Entities

Aside from studying the effect of varying radius on the moving flock algorithm, the effect of changing the order of entities in the input file was also investigated. For each of the semi-synthetic DNP, the Fontainebleau and the Delft datasets, 4 versions of the same datasets were obtained by changing the order of the entities compared to the original. Then, the moving flock algorithm was run 15 times for each of the original datasets and the varied versions. The same set of parameter values as shown in Table 4.8 were used for a dataset and its reordered versions.

Dataset	min_points	min_time_slices	synchronization_rate	radius
Semi-synthetic DNP	3	3	300s	150m
Fontainebleau	3	3	60s	150m
Delft	3	3	60s	50m

Table 4.8: The set of parameter values used for the datasets.

Table 4.9 summarizes the differences between the obtained flocking results from the three original datasets and their corresponding reordered versions. *Flock Count in Original* refers to the total number of flocks found using the original dataset. The succeeding columns give a count of the differences between the results obtained from the original dataset compared to its reordered version. A minor change refers to slight changes in the time and extent of flocking while a major change either refers to a missed flock (found in DNP) or a change in the flock membership composition (found in Delft). The DNP dataset has more or less 1 minor and 1 major differences compared to its reordered versions. As for the Fontainebleau dataset, there was no variation in the flock results. Finally, the Delft dataset has 2-3 differences compared to other versions.

	Flock Count in Original	Version 1	Version 2	Version 3	Version 4
Semi-synthetic DNP	11	1 minor	1 minor; 1 major	1 minor; 1 major	1 minor; 1 major
Fontainebleau	2	0	0	0	0
Delft	10	2 minor; 1 major	1 minor; 1 major	2 minor; 1 major	2 minor

Table 4.9: Difference between the original datasets and their corresponding reordered versions.

For further understanding of these differences, Table 4.10 provides specific examples of minor and major differences in the DNP and the Delft datasets. Example 1 is an instance of a minor difference in the DNP dataset. A shorter flocking duration and thus, a shorter extent was obtained from Version 1. Example 2 is an instance of a major difference wherein the flock discovered from DNP was not found from its Version 2. Example 3 shows a minor difference, wherein the discovered flocks only vary in the end time, for Delft. A major difference wherein the flock members in the discovered flock varies is given in Example 4. Object 212 was replaced by object 715 in the flock obtained from Delft’s Version 3. The start time and flock extent also varied consequently.

	<i>Datasets</i>	<i>Difference Type</i>	<i>Start Time</i>	<i>End Time</i>	<i>Flock Extent</i>	<i>Flock Members</i>
1	Semi-synthetic DNP and Version 1	Minor	Thu Dec 30 09:40:00 1999	Thu Dec 30 10:00:00 1999	628.5	52; 100; 46
			Thu Dec 30 09:45:00 1999	Thu Dec 30 09:55:00 1999	279.125	46; 52; 100
2	Semi-synthetic DNP and Version 2	Major	Thu Dec 30 12:40:00 1999	Thu Dec 30 12:50:00 1999	178.5	139; 365; 140
			Not found			
3	Delft and Version 2	Minor	Thu Nov 19 10:50:00 2009	Thu Nov 19 11:00:00 2009	55.5	223; 708; 221
			Thu Nov 19 10:50:00 2009	Thu Nov 19 11:05:00 2009	55.5	221; 223; 708
4	Delft and Version 3	Major	Thu Nov 19 10:45:00 2009	Thu Nov 19 10:50:00 2009	70.5	212; 708; 203
			Thu Nov 19 10:46:00 2009	Thu Nov 19 10:50:00 2009	52	203; 715; 708

Table 4.10: Difference between the results obtained from the original datasets and their corresponding reordered versions.

In general, the differences in the discovered flocks are due to the different bases used for each version. Since sub-trajectories already included in previously extracted moving flocks are filtered out, they cannot be used as base trajectories in finding other flocks. This technique greatly helped in filtering out redundant flocks but also caused slight changes in the obtained flock results when the object entries are reordered.

4.2.5 Validation of the Moving Flock Algorithm

This part provides a discussion of how the moving flock algorithm is validated using the null hypothesis principle. The aim is to show that the obtained flocks are inherent in the input and not obtained by mere chance. We used two randomization techniques and obtained different versions of the input dataset. Afterwards, the flock algorithm was run on different versions of the input dataset and their results were compared.

We split the discussion of the validation experiment into two parts, one for randomization by using Markov chain and the other for randomization based on uncertainties in collected spatial points.

Validation through Markov Chain Randomized Datasets

We have randomized the semi-synthetic DNP, the Fontainebleau and the Delft dataset by building a Markov chain for each based on their underlying data distribution of spatial points. As with the parameters used in observing the effect of ordering on the algorithm, the same parameter values as shown in Table 4.8 of Section 4.2.4 were used for the datasets and their randomized versions.

The randomization algorithm was ran several times for each dataset. In the case of DNP, it was randomized for around 10 times and only 4 of the randomized datasets yielded some flocks. There was 1 flock obtained from 3 of the randomized datasets, and there were

6 flocks extracted from the other randomized dataset. Recall that 11 flocks were found in the original dataset. Aside from varying in the number of flocks discovered, the properties of the flocks themselves are also very different as shown in Table 4.11.

	<i>Start Time</i>	<i>End Time</i>	<i>Flock Extent</i>	<i>Flock Members</i>
Original	Thu Dec 30 12:15:00 1999	Thu Dec 30 12:30:00 1999	991.9375	96; 288; 15
	Thu Dec 30 09:40:00 1999	Thu Dec 30 09:55:00 1999	870.5	228; 287; 104
	Thu Dec 30 11:55:00 1999	Thu Dec 30 12:05:00 1999	692.4375	118; 249; 346
	Thu Dec 30 09:40:00 1999	Thu Dec 30 10:00:00 1999	628.5	52; 100; 46
	Thu Dec 30 11:40:00 1999	Thu Dec 30 11:50:00 1999	472.1875	38; 349; 223
	Thu Dec 30 11:15:00 1999	Thu Dec 30 11:45:00 1999	432.5	113; 215; 112
	Thu Dec 30 10:35:00 1999	Thu Dec 30 10:45:00 1999	269.3125	303; 342; 38
	Thu Dec 30 11:20:00 1999	Thu Dec 30 11:30:00 1999	209	158; 203; 78
	Thu Dec 30 12:45:00 1999	Thu Dec 30 13:00:00 1999	203.9375	139; 141; 140
	Thu Dec 30 12:40:00 1999	Thu Dec 30 12:50:00 1999	178.5	139; 365; 140
	Thu Dec 30 12:50:00 1999	Thu Dec 30 13:00:00 1999	150.125	140; 139; 142; 129
Version 1	Thu Dec 30 11:50:00 1999	Thu Dec 30 12:00:00 1999	179.249846329912	180; 315; 40
Version 2	Thu Dec 30 11:20:00 1999	Thu Dec 30 11:30:00 1999	178.807183507829	253; 337; 42
Version 3	Thu Dec 30 11:10:00 1999	Thu Dec 30 11:20:00 1999	402.023812355706	103; 155; 57
	Thu Dec 30 11:00:00 1999	Thu Dec 30 11:10:00 1999	221.557437454815	93; 289; 115
	Thu Dec 30 11:55:00 1999	Thu Dec 30 12:05:00 1999	212.980809040018	38; 332; 118
	Thu Dec 30 10:30:00 1999	Thu Dec 30 10:40:00 1999	209.968162752571	57; 118; 115
	Thu Dec 30 11:40:00 1999	Thu Dec 30 11:50:00 1999	189.306787050329	204; 252; 343
	Thu Dec 30 10:15:00 1999	Thu Dec 30 10:25:00 1999	166.101779716089	2; 57; 336
Version 4	Thu Dec 30 12:45:00 1999	Thu Dec 30 12:55:00 1999	336.722883264767	41; 264; 192

Table 4.11: Moving flock results for different randomized versions of the semi-synthetic DNP dataset.

The randomization algorithm was also run several times on the Fontainebleau dataset but after several runs, no moving flock was found in any of these randomized versions while there were 2 flocks found in the original. The problem with this dataset is that there were a few noises included in it that caused the Markov chain to generate random values restricted by the minimum and maximum bounds of the trajectories. As a consequence, the dataset was randomized with minimal constraint, making the points quite random and making it difficult to form flocks in the randomized dataset.

The Delft dataset was randomized 4 times. While 10 flocks were obtained from the original dataset, 16, 18, 22, and 27 flocks were extracted from the randomized datasets. This means that certain grid movements have high probability in the Markov chain and causes the creation of arbitrary flocks in the randomized datasets. The flock results obtained from the original and the random datasets had some flock members in common but they are grouped together in different ways. In the few cases that the flocks from the different versions have 2 members in common, they are still dissimilar based on the variation in the time of flocking and thus, the extent as well. There was only 1 case wherein 2 flocks with overlapping time duration and with 2 common members were found from 2 randomized versions of the dataset. The properties of these flocks are shown in Table 4.12.

<i>Start Time</i>	<i>End Time</i>	<i>Flock Extent</i>	<i>Flock Members</i>
Thu Nov 19 12:31:00 2009	Thu Nov 19 12:33:00 2009	84.9541098016779	214; 718; 734
Thu Nov 19 12:32:00 2009	Thu Nov 19 12:34:00 2009	57.3686375541146	718; 734; 212

Table 4.12: Two moving flocks obtained from two randomized versions of Delft.

Validation through Uncertainty-based Randomized Datasets

A subset of the semi-synthetic DNP dataset, which is similar to the one used in investigating the radius' effect on the algorithm's results, was used in this set of experiments. 6 randomized versions of this dataset were generated by replacing (x, y) pairs in the original

dataset with a new value bounded by a user-specified uncertainty radius. The generated dataset is more distorted when the radius value is larger.

Table 4.13 presents the extracted flocks from each of the datasets randomized with varying radius values. The results are sorted by decreasing radius values, starting with the more distorted dataset to the less distorted dataset. As the radius value becomes smaller, it is expected that the flock results also becomes more similar to those extracted from the original dataset. This is verified by the results found in the table. The moving flock algorithm was run on all datasets, including the original and randomized versions, with the same parameters: $min_points=3$, $min_time_slices=3$, $synchronization_rate=300s$, and $radius=150m$.

	<i>Start Time</i>	<i>End Time</i>	<i>Flock Extent</i>	<i>Flock Members</i>
Original	Thu Dec 30 12:15:00 1999	Thu Dec 30 12:30:00 1999	991.9375	96; 288; 15
	Thu Dec 30 12:45:00 1999	Thu Dec 30 13:00:00 1999	203.9375	139; 140; 141
	Thu Dec 30 12:50:00 1999	Thu Dec 30 13:00:00 1999	150.125	139; 140; 129
Version 1 (100m)	Thu Dec 30 12:05:00 1999	Thu Dec 30 12:20:00 1999	219.589825248345	147; 85; 125
	Thu Dec 30 12:05:00 1999	Thu Dec 30 12:15:00 1999	151.693097695591	125; 85; 23
Version 2 (50m)	Thu Dec 30 12:15:00 1999	Thu Dec 30 12:25:00 1999	636.779477782431	96; 288; 15
	Thu Dec 30 12:45:00 1999	Thu Dec 30 12:55:00 1999	265.550003897515	140; 141; 139
	Thu Dec 30 12:35:00 1999	Thu Dec 30 12:50:00 1999	162.296600496396	223; 264; 217
Version 3 (40m)	Thu Dec 30 12:20:00 1999	Thu Dec 30 12:30:00 1999	699.585639138706	288; 96; 15
	Thu Dec 30 12:45:00 1999	Thu Dec 30 13:00:00 1999	225.666424336261	140; 141; 139
	Thu Dec 30 12:40:00 1999	Thu Dec 30 12:50:00 1999	197.100121075171	140; 365; 139
Version 4 (30m)	Thu Dec 30 12:15:00 1999	Thu Dec 30 12:30:00 1999	1038.6139229855	288; 96; 15
	Thu Dec 30 12:45:00 1999	Thu Dec 30 13:00:00 1999	202.910822035046	140; 141; 139
Version 5 (20m)	Thu Dec 30 12:15:00 1999	Thu Dec 30 12:30:00 1999	1015.33351584302	288; 96; 15
	Thu Dec 30 12:40:00 1999	Thu Dec 30 12:50:00 1999	211.040869345073	140; 365; 139
	Thu Dec 30 12:45:00 1999	Thu Dec 30 13:00:00 1999	193.683468844508	140; 141; 139
Version 6 (10m)	Thu Dec 30 12:15:00 1999	Thu Dec 30 12:30:00 1999	990.175165907945	288; 96; 15
	Thu Dec 30 12:45:00 1999	Thu Dec 30 12:55:00 1999	198.080599969602	140; 141; 139
	Thu Dec 30 12:50:00 1999	Thu Dec 30 13:00:00 1999	159.25492139568	129; 140; 139

Table 4.13: Moving flock results for different randomized versions of a subset of the semi-synthetic DNP dataset.

The result of this investigation shows that different flocks are obtained when the dataset is randomized by a large enough radius, like 100m in this case. Therefore, the obtained moving flocks depends on the dataset and are not discovered by chance. Moreover, as the radius threshold of the randomization algorithm is decreased, the extracted flocks become more similar to those found in the original dataset. This demonstrates the robustness of the flocking algorithm to uncertainties in the observation points of the input dataset.

4.2.6 Summary of Results

Compared to existing flock detection methods, we introduce the concept of moving flocks along with a method for computing such patterns. This concept emphasizes that flock members should be moving, disqualifying patterns with flock members that remain stationary in a common place during the considered time duration. Moreover, the algorithm allows the end user to find moving flock patterns in pedestrian movement as demonstrated in the discussed results. One main research finding is related to the tendency of visitors to flock when following certain paths provided in the recreational areas, an example of which is the White route followed by certain flocks in DNP.

We have considered the concept of spatio-temporal coherence, recently introduced by Wood and Galton in [92] for defining collectives, since it is a behavior exhibited by members of a moving flock. The flock consisting of several members is a collective of objects exhibiting spatial closeness over some time duration with a minimum number of members. Thus, one of the main spatio-temporal coherence criteria is the radius since

it defines the closeness of visitors of a moving flock over time. Whereas it is intuitive to discover more moving flocks as the radius is set to a larger value, this is not always the case. It is possible to find a smaller number of moving flocks with a larger value for the radius since the spatial extent constraint also becomes more restrictive for large values of radius. A set of guidelines was described in selecting appropriate values for the algorithm's parameters. A specific technique for finding an appropriate radius value was proposed in subsection 3.1.3.

The moving flock algorithm was tested on different tracking datasets, where the trajectories of pedestrians in different types of recreational landscapes were collected from GPS devices. The results have shown that the discovered patterns have varying time durations and spatial extents, as well as that many of them were located at the most popular routes in both parks.

The experiments also revealed the filtering power of the proposed algorithm. By sorting the results using the spatial extent of flocks and removing those with very short extent, we have pruned out a large number of uninteresting patterns. These patterns include both the stationary flocks and the redundant patterns, which should not be considered as collectives of objects moving together. However, it is also important to point out that some redundant patterns have been retained during the analysis in order to avoid losing moving flock patterns with longest durations.

Since the approximation algorithm does not consider all possible points as center (i.e., it only considers the points found in the dataset as center), it may not compute some interesting flock patterns. In approximating the flocking results, the algorithm assumes that if a pedestrian is located at the center of the flock during starting time instance, then the same pedestrian remains at the center of the flock for the rest of the time instances. These issues can be resolved at the expense of a higher running time. For instance, instead of only recursively merging 'basic flocks' obtained using the same base trajectory, those that were obtained using different base trajectories can be merged as well.

We have also considered comparing actual performances of existing algorithms compared with our implementation. However, the implementations of most algorithms are either non-existing or unavailable. The moving cluster algorithm was made available to us but it is designed for finding flocks with varying members rather than with fixed members as in our case. Thus, we can only provide a comparison at conceptual level (recall Table 2.1) rather than at performance level among the algorithms.

The algorithm has been validated to extract results inherent in the input dataset. In addition to this, the experiments demonstrate that the algorithm is quite robust with respect to uncertainties in the collection of the recorded points and to changes in the ordering of the entities in the input. The accuracy of the GPS receivers used in both datasets is around 3m and it has been empirically shown that the flocking results obtained from the true observation points are very similar to those obtained from observations with small inaccuracies.

An implementation of the moving flock discovery algorithm is available at the following URL: <http://www-kdd.isti.cnr.it/moving-flock>.

4.3 Semantic Annotation

Two of the datasets we used for testing the moving flock discovery algorithm contain the pedestrians’ responses to surveys conducted in their respective recreational areas. These are the DNP and the Delft dataset.

A current challenge is that most datasets do not explicitly contain semantic information needed for the flock interpretation step. In the case that they do contain these information, these information usually require cleaning to minimize errors present in the data and there is currently no formal approach for assessing the accuracy of semantic data. Furthermore, datasets in general are most likely kept by a private organization who have some hesitation in releasing them. However, with further advancement in location and privacy-aware technologies, such datasets can be made more available in the future.

4.3.1 DNP

This section provides a description of the use of Wood and Galton’s taxonomy [92] for the selection of the semantic attributes, and the semantic annotation step performed at individual and flock levels. The discussion for both levels also includes how the semantic attributes for annotation were selected. Recall that these attributes were obtained from visitors’ responses to conducted surveys in the park as described in Section 4.1.

Semantic Attribute Selection

Using Wood and Galton’s taxonomy [92], we have mapped the candidate semantic attributes to their corresponding classification criteria as shown in Table 4.14. Some attributes, such as *Visitor type*, can be mapped to more than 1 criterion. For the purpose of attribute selection, mapping to all possible criterion is not important. It is enough to map the attribute to at least one criterion. Attributes that were not mapped to any criterion are disregarded for semantic annotation and are no longer shown in the table.

The attributes found in the table are further filtered by removing redundant and unary attributes. For example, *Is a dogwalker?* can be derived from *Dog number* and is thus, disregarded. Another example is *Currant forests*, which was also disregarded since it has a unary value of 0 (i.e., false) among flock members.

Semantic Annotation in DNP at Individual Level

As mentioned earlier, we propose two levels of annotation, which are the individual and the flock level. Both were applied in annotating the DNP dataset.

At the individual level, individual trajectories belonging to discovered flocks were annotated with their corresponding visitor characteristics. An example of a visitor characteristic is *visitor.type*, which specifies whether the visitor is an elderly person, an adult, an elderly couple, an adult couple, a family with children, a group of adults, or a family consisting of adults. Aside from the visitor characteristics, each trajectory was also annotated with a set of flock membership attributes, indicating whether the visitor that made the trajectory belongs to a specific flock or not. These make up the set of individual attributes.

Figure 4.13 shows a sample semantic annotation of 3 flock members in the DNP dataset. *r_id* refers to the entity ID, *on_holiday* is a boolean value describing whether the visitor is

<i>Criteria</i>	<i>Semantic Attributes</i>
<i>Membership</i>	Is in the area for holiday? Frequency of visit. Visited since when. Total number of visited attractions. Number of information sources used. Is a local? Has visited an attraction? Is a browser? Is a repeater? Is a dogwalker? Is with children? Age category
<i>Location</i>	Has visited the: [Picnic areas, Mound, Currant forests, Information centre, Woods, Bird watching sites, Prayer areas, Juniper berries, Fens, Sheepfold areas, Snack bar areas, Sightseeing areas, Radio telescope, David lakes, Orienting, Teahouses]? Followed a route? Has stopped? Type of stop. Stopped for: [Catering, Beautiful, Quiet, Seat, Lunch]? Has followed the: [white route, whiteLheederzand, redSpier, blue route, redLheederzand, yellowLheederzand, redLheederzandEast, redDiepveen, yellowLheebroekerzand, whiteSpier, blueSpier, whiteVC, redVC, Drenthepad, brochureVC, brochureSpier, brochureVijfsprong, kompasrouteVC, brochureDiepveen, brochureLheederzand, greenSpier]?
<i>Coherence</i>	Main activity. Visiting purpose. Visting purpose type. Visiting goal. Is attracted to the park due to: [Parking accessibility, Sheepfold proximity, Catering proximity, Quietness, Attraction proximity, Route start, Coincidence]?
<i>Roles</i>	
<i>Depth</i>	Adult number. Children number. Visitor type. Dog number. Total number of persons

Table 4.14: Mapping of semantic attributes to Wood and Galton’s criteria.

in the area for a holiday or not, *freq_visit* describes how often the visitor comes to the park, *adult_num* is the number of adults represented by the current entity, and *children_num* is the number of children included in the current entity. The attributes *adult_num* and *children_num* specifically implies that each entity in the dataset may consist of a group of adults and/or children. Meanwhile, *picnic_areas*, *mound*, *bird_watching_site*, and *prayer_areas* describe whether the entity visited these attractions or not. Finally, *flock0* and *flock1* are attributes generated by the flock discovery algorithm and they indicate whether the entities belong to any specific flock or not.

r_id character varying(10)	on_holiday integer	freq_visit integer	adult_num integer	children_num integer	picnic_areas integer	mound integer	bird_watching_site integer	prayer_areas integer	flock0 integer	flock1 integer
R195	1	2	2	0	0	0	0	0	1	0
R647	0	1	2	0	0	0	0	0	1	0
R015	0	3	2	0	0	0	1	0	1	0

Figure 4.13: Semantic Annotation of Individual Flock Members in DNP.

Out of the 84 individual semantic attributes, which consists of 73 survey-based attributes and 11 flock membership attributes, 51 attributes were used for the semantic annotation step. With the help of the domain expert, 19 survey-based attributes were deemed as unnecessary or redundant. *gps_true*, which indicates whether the gps entry is valid or not, was considered as an unnecessary attribute since the gps entries in the dataset were already preprocessed and cleaned. An example of a redundant attribute is *walking_with_children*, which indicates if a visitor is accompanied by children. This boolean information can be derived from *children_num*, which specifies the number of accompanying children.

3 unary attributes, which only contains 1 value for all flock members, were automati-

cally removed using WEKA's *RemoveUseless* filter. These attributes include visiting the current forest, visiting the woods, and the importance of quietness.

Moreover, 11 attributes whose values are only either 0 (corresponds to false) or 99 (corresponds to null) for all flock members were also manually removed since these values do not give very interesting information. An example of this is *blue_route*, which means that some members did not follow the route while it is not known whether others have followed this route or not.

Semantic Annotation in DNP at Flock Level

In addition to annotating individual trajectories of flock members, flocks themselves are also semantically annotated. In this case, the set of flock properties is composed of the parameters used by the flock detection algorithm, the generated flock descriptions and the aggregated properties of individual flock members. The parameters used to discover flock patterns include *min_points*, *radius*, *min_time_slices*, and *synchronization_rate*, as described in Section 3.1.1. Examples of flock descriptions provided by the flock discovery algorithm are the start and end time of flocking, the ID of the flock members, the spatial extent covered by the flock, the duration of flocking, the number of flock members, the average speed of flock members, and others. Some examples of the aggregated properties on the visitor type are *visitor_type_1* (i.e., elderly alone), *visitor_type_2* (i.e., adult alone), *visitor_type_3* (i.e., elderly couple), etc. These properties are extracted from the individual property *visitor_type*. This attribute can have the following values: 1, 2, 3, 4, 5, 6, 7, 99 (i.e., unknown). An aggregated property is created for each of these possible values, thus, producing eight aggregated properties. The value of each aggregated property depends on the number of individuals satisfying the considered individual attribute value. For example, if a given flock pattern has 1 out of 3 members whose *visitor_type* attribute is equal to 1, then the flock property *visitor_type_1* of the flock is set to 0.33 (i.e., 1/3).

Figure 4.14 shows a subset of the semantic attributes at flock level upon applying the framework on the DNP dataset. In this example, there are 10 discovered flocks, each one having the *on_holiday* and the *freq_visit* aggregated attributes. *on_holiday* at the individual level contains 3 possible values: 0, 1, null. Thus, at the flock level, there are 3 attributes associated with it. Likewise for the *freq_visit*, it has 6 possible values: neg, 1, 2, 3, 4, 5 and hence, there are 6 corresponding attributes at the flock level. These attributes describe the percentage of flock members having the specified value for the considered individual level attribute. For instance, *on_holiday_0* for flock 0 has a value of 0.666667 indicating that 66.6667% of flock 0's members have a value of 0 for the *on_holiday* attribute. On the other, *on_holiday_1* has a value of 0.333333, which means that 33.3333% of flock 0's members have a value of 1 for the *on_holiday* attribute. In layman's term, 33.3333% of flock 0's members are in the area for a holiday while the remaining percentage are not.

A total of 108 attributes, which include survey-based properties and algorithm generated descriptions such as *start_time* of flocking, were selected for flock level annotation. The survey-based attributes were based on the selected individual level attributes.

The parameters used to extract the moving flocks were considered as unnecessary since the flocks considered were obtained using exactly the same parameters. In other words, the attributes obtained from the parameters have unary values. Therefore, they were disregarded as flock level attributes.

flock_id integer	on_holiday_0 real	on_holiday_1 real	on_holiday_null real	freq_visit_neg real	freq_visit_1 real	freq_visit_2 real	freq_visit_3 real	freq_visit_4 real	freq_visit_5 real
0	0.666667	0.333333	0	0	0.333333	0.333333	0.333333	0	0
1	0.333333	0.666667	0	0.333333	0.333333	0	0	0	0.333333
2	0.333333	0.666667	0	0	0.666667	0.333333	0	0	0
3	0.666667	0.333333	0	0	0.333333	0	0.666667	0	0
4	0.333333	0.666667	0	0	0.333333	0.333333	0.333333	0	0
5	0.333333	0.666667	0	0	0.333333	0	0.333333	0.333333	0
6	1	0	0	0	0.666667	0	0.333333	0	0
7	0.666667	0	0.333333	0	0	0	1	0	0
8	0.333333	0.666667	0	0	0.333333	0	0.666667	0	0
9	0.333333	0.666667	0	0	0.333333	0	0.666667	0	0
10	0.5	0.5	0	0	0.25	0	0.75	0	0

Figure 4.14: Semantic Annotation of Discovered Flocks in DNP.

Flock generated attributes that were removed include *base_id* and *min_speed*. *base_id* uniquely identifies the base trajectory used to find the flock while *min_speed* refers to the minimum speed of the base trajectory during the time of flocking. Both were automatically removed using WEKA’s *RemoveUseless* filter since their values vary too much.

Moreover, pairs of complementary survey-based flock attributes were considered to find redundant attributes to be removed. For example, the complement of *bird_watching_site_0* is *bird_watching_site_1* and, vice versa since the value of an attribute can be easily computed from the other. Thus, one of these attributes may be removed. Recall that *bird_watching_site_0* is the percentage of flock members who did not visit the bird watching site, while *bird_watching_site_1* is the percentage of those who did.

4.3.2 Delft

The same approach to semantic annotation was also applied to the Delft dataset. In this case, however, we did not have access to the questionnaire used in conducting the survey but we were provided with the pedestrian attributes obtained from the survey responses.

This section describes the annotation performed at individual and flock level, which includes the selection of the semantic attributes used for this phase of the framework.

Semantic Annotation in Delft at Individual Level

Compared to the DNP dataset, Delft has a fewer set of semantic attributes. There are 39 attributes all in all, and this includes the 29 survey-based attributes and the 10 flock membership attributes. The survey-based attributes include information about the the pedestrian’s age, gender, interest in shopping, postal codes, the weather condition, and others.

12 of the survey-based attributes were deemed as unnecessary or redundant. For example, *postcodea* and *postcodeb* both refer to zip codes that come in different formats. Thus, it is enough to use only one of them. Another example is the GPS device ID, which was considered unnecessary since we are not interested in associating people with the GPS device they used.

Considering only the semantic attributes of flock members, *destination* only had a unary value of 2 and hence, was disregarded to simplify the analysis task. This means that all pedestrians involved in flocking were headed for destination 2.

As with the DNP dataset, only trajectories of flock members were annotated with their survey-based and flock membership attributes. Figure 4.15 shows a subset of the semantic annotation step performed on 3 flock members extracted from Delft. Note that this subset does not include all attributes used for annotation. *id* uniquely identifies each pedestrian, *purpose* describes the pedestrian’s purpose in going downtown, *shopping* indicates the type of shopping performed, *postcodeb* refers the pedestrian’s postal code, *occup* refers to his/her occupation, and *wth_sunny* indicates the pedestrian’s preference of sunning weather when going downtown. *gender* has 2 possible values equivalent to being male or female while *age* refers to the current age of the pedestrian. *flock0*, *flock1*, and *flock2* are flock membership attributes and they indicate whether the pedestrian belongs to flock 0, flock 1 and/or flock 2, respectively.

id integer	purpose integer	shopping integer	postcodeb character varying(10)	gender integer	age integer	occup character varying(300)	wth_sunny integer	flock0 integer	flock1 integer	flock2 integer
203	1	3	EN	2	55	Housewife	-1	0	1	0
212	4	3	MH	2	52	Housewife	-1	0	1	0
708	1	3	VH	1	41	Housewife	-1	0	1	0

Figure 4.15: Semantic Annotation of Individual Flock Members in Delft.

Semantic Annotation in Delft at Flock Level

As with the DNP dataset, the set of flock properties is composed of the parameters used by the flock detection algorithm, the generated flock descriptions and the aggregated properties of individual flock members. Again, the parameters are disregarded since we are dealing with flocks obtained using the same set of parameter values. The remaining attributes generated by the flock algorithm are *base_id*, *num_of_time_slices* and *ave_speed* since they others were either deemed as unnecessary, redundant or unary attributes. As for the survey-based attributes, they were selected based on the chosen individual survey-based attributes and one of the pair of complementary attributes were also filtered out. An example of a complementary pair of attributes is *gender_1* and *gender_2*. The attribute value for one of them can be derived from the other since a pedestrian of female gender can be inferred from the male gender attribute by using the negation operator. We chose to arbitrarily remove *gender_2*.

Figure 4.16 provides a sample of the semantic annotation of the 10 flocks extracted from the Delft dataset. The individual attribute *purpose* has 4 possible values and thus, there are 4 corresponding flock level attributes for it. Likewise, there are 4 possible values for the individual attribute *shopping*, giving rise to 4 corresponding flock level attributes.

4.4 Pattern Analysis

This section describes how the proposed framework can be used to interpret the moving flock patterns discovered in the DNP and the Delft dataset once the semantic annotation step has been completed. The last step of the framework, the pattern analysis step, includes the execution of selected data mining tasks on the individual attributes, the flock attributes and the discovered flocks. Specifically, it includes correlation computation and hierarchical clustering, and classification.

	flock_id integer	purpose_1 real	purpose_2 real	purpose_3 real	purpose_4 real	shopping_0 real	shopping_1 real	shopping_2 real	shopping_3 real
1	0	0.666667	0.333333	0	0	0	0	0.333333	0.666667
2	1	0.666667	0	0	0.333333	0	0	0	1
3	2	0.666667	0	0.333333	0	0	0	0.333333	0.666667
4	3	0.666667	0	0	0.333333	0	0	0	1
5	4	1	0	0	0	0	0	0.333333	0.666667
6	5	1	0	0	0	0	0	0.333333	0.666667
7	6	0.666667	0	0.333333	0	0	0.333333	0.333333	0.333333
8	7	1	0	0	0	0	0	0.333333	0.666667
9	8	1	0	0	0	0	0	0	1
10	9	0.666667	0	0	0.333333	0.333333	0	0.666667	0

Figure 4.16: Semantic Annotation of Discovered Flocks in Delft.

SUC (Symmetrical Uncertainty Coefficient) [11] and the Pearson’s correlation coefficient [21] were used to compute the correlations among the individual and the flock attributes, while a flock similarity measure described in Section 3.3.1 is used for computing the distance scores among the discovered flocks.

An ensemble of Java codes using WEKA [12] classes and R [17] codes were used to perform the pattern analysis step. Some WEKA classes were used to perform additional preprocessing steps and to compute the symmetrical uncertainty coefficient. R’s built-in functions were used to compute Pearson’s correlation coefficient and to perform hierarchical clustering. Meanwhile, a cost-sensitive implementation of J48 in WEKA was utilized in obtaining the classification results.

A brief discussion of the flock interpretation results that are specific to each dataset are found in the succeeding sections.

4.4.1 DNP

We focused on the 11 moving flock patterns that were extracted from the semi-synthetic version of the dataset using the following parameters: $min_points = 3$, $radius = 150m$, $min_time_slices = 3$, $synchronization_rate = 300s$. Each flock has 3-4 members each, and the members remain spatially close for 3-7 time instances (i.e., 10-30 minutes).

It is important to recall that the moving flocks in this experiment were discovered from the collapsed dataset and thus, these flocks are actually occurring at the same time of different days. The obtained interpretations from this experiment describe the common properties/behaviors of people who follow the same route on the same time of possibly different days.

Correlation Computation and Hierarchical Clustering Results

Analyzing Individual Attributes of Flock Members This section describes the results obtained from performing the correlation computation and the hierarchical clustering steps when considering only trajectories belonging to discovered flocks. Doing so allows the discovery of relations that only exist among flocking individuals. Moreover, time and effort are saved since analysis is only concentrated on flock members.

The obtained results were encouraging as we have found interesting relations among individual attributes despite of the fact that there were only 29 trajectories involved in flocking out of the 370 trajectories available in the dataset. The most interesting relations are those that are between a flock membership attribute and a visitor characteristic

attribute since this type of relations allows the user to understand the common characteristics that possibly cause them to flock together.

We have used SUC, Pearson’s correlation coefficient, and the absolute value of Pearson’s correlation coefficient for correlation computation. The computed correlations are then used to hierarchically cluster the attributes. The dendrograms obtained using the different correlation measures are consistent (i.e., the groupings of attributes are quite similar though their scores and ranking may vary slightly based on visual inspection) and thus, we only describe a subset of the discovered relations found using the standard correlation coefficient, which allows the analyst to concentrate on positive correlations. Figure 4.17 shows the obtained tree, focusing on a sample of interesting relations that were found. The first set of relations labelled **a)** can be split further into 2 groups: one involving *adult_num* and *visitor_type*, and the other involving the rest of the attributes. The first group validates that the relations obtained are inherent in the dataset since visitor type is defined in terms of the number of adults and the number of childrens. The second group of relations indicate that belonging to *flock5* is linked to the visitor’s interest in sheepfold attractions. Furthermore, this group also exhibits the validity of the obtained results by finding a trivial relation between *sheepfold_proximity* (i.e., attracted to the park due to its proximity to sheepfold areas) and *sheepfold_areas* (i.e., visited the sheepfold areas). The next set of relations labelled **b)** indicates that *flock8*, *flock9* and *flock10* are closely related, and they are linked to the White route of the park. Indeed, these 3 flocks have members in common and the White route was described by the domain expert as the most popular route in the park. Meanwhile, set **c)** shows that belonging to *flock2* is closely related to the member’s preference of visiting mounds and bird watching sites. Finally, set **d)** describes how visitors belonging to *flock0* are linked with *coincidence*, which means they are attracted to the park for no specific reasons. In addition to this, they are also linked to 4 attractions, which includes radio telescope, david lakes, juniper berries and fens. Sets **b)**, **c)** and **d)** can be explained further using classification analysis.

Analyzing Flock Attributes Analysis of the flock attributes is important as well since it shows how the correlations among the visitor characteristics change when only individual flock members are considered and when the entire flock as a collective is considered. Some of the most interesting relations among flock attributes are those related to age category, visitor type, and main activity.

Since the flock attributes are all numeric, we used both SUC and the two versions of Pearson’s correlation coefficient to compute the correlation scores as was done for the individual attributes. The clustering results derived using the three different correlation measures are consistent. Therefore, we only present some examples of the result found in Figure 4.18 using Pearson’s correlation measure. The relation labelled **a)** is a trivial relation known to the domain expert since *visitor_type_4* corresponds to the percentage of adult couples in the flock while *children_num_0* are the percentage of members without any children. Finding these types of relations validates the credibility of other less obvious relations. Relation **b)** indicates that if more members belong to the 2nd age group, which corresponds to the age range 30-59, then more members have visited the bird watching site as well. Meanwhile, relation **c)** indicates that more members classified as visitor type 3, which refers to an elderly couple, implies having more members involved in main activity 8, which refers to “others” and can be interpreted as unpopular activities in the park. In

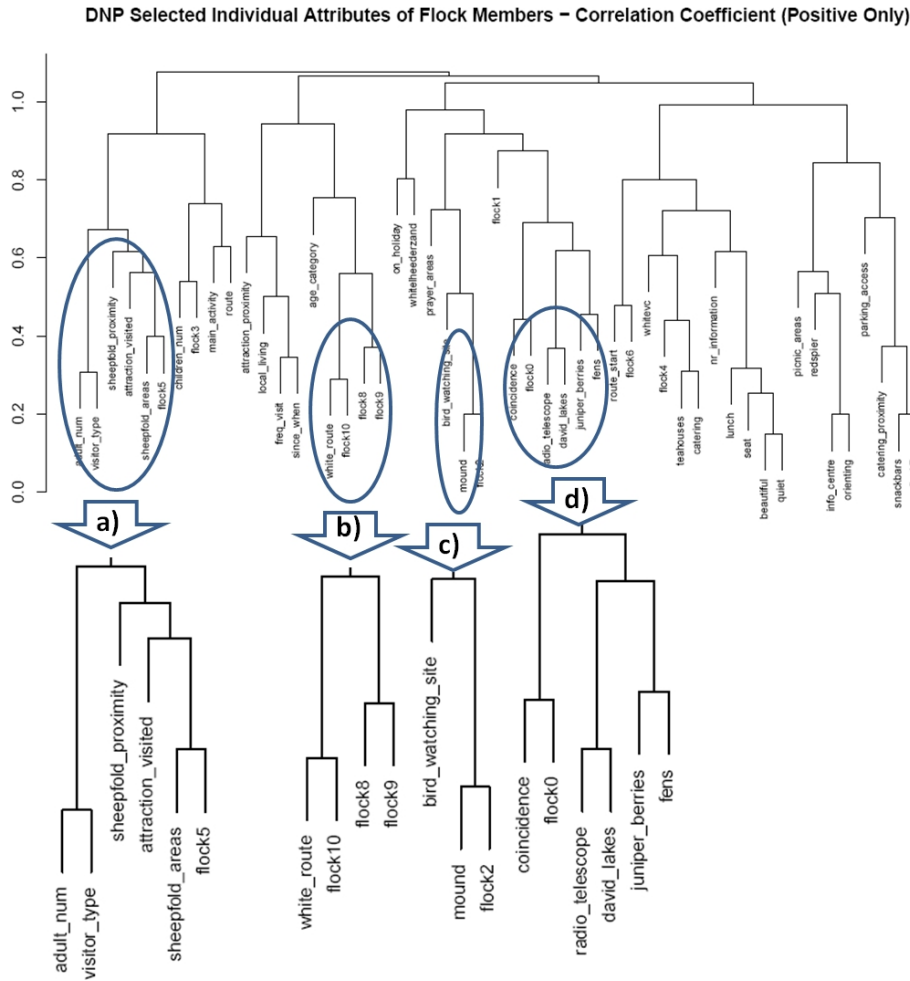


Figure 4.17: Sample relations among individual attributes of flock members in semi-synthetic DNP.

other words, the relation leads to the generalization that most elderly couples are involved in the less popular activities of the park. Relation **d**) involves a survey-based and a flock algorithm generated attribute. It states that the number of flock members is related to the frequency of visitors following the White route. Finally, relation **e**) involves two attributes generated by the flock algorithm. This relation is expected, since flocking that occurs at an earlier starting time tend to end earlier. These sample relations illustrate different types of relations that can be found: among survey-based attributes, flock algorithm generated attributes, or a combination of both.

Analyzing Flock Entries The flock entries extracted during the pattern discovery step are also clustered in order to understand which flocks are more similar to each other. Combined with the clustering results for the individual attributes, it is possible to pinpoint specific attributes that make flocks similar to other flocks.

Using the un-weighted and the weighted distance measures described in Section 3.3.1 for hierarchical clustering gave similar results.

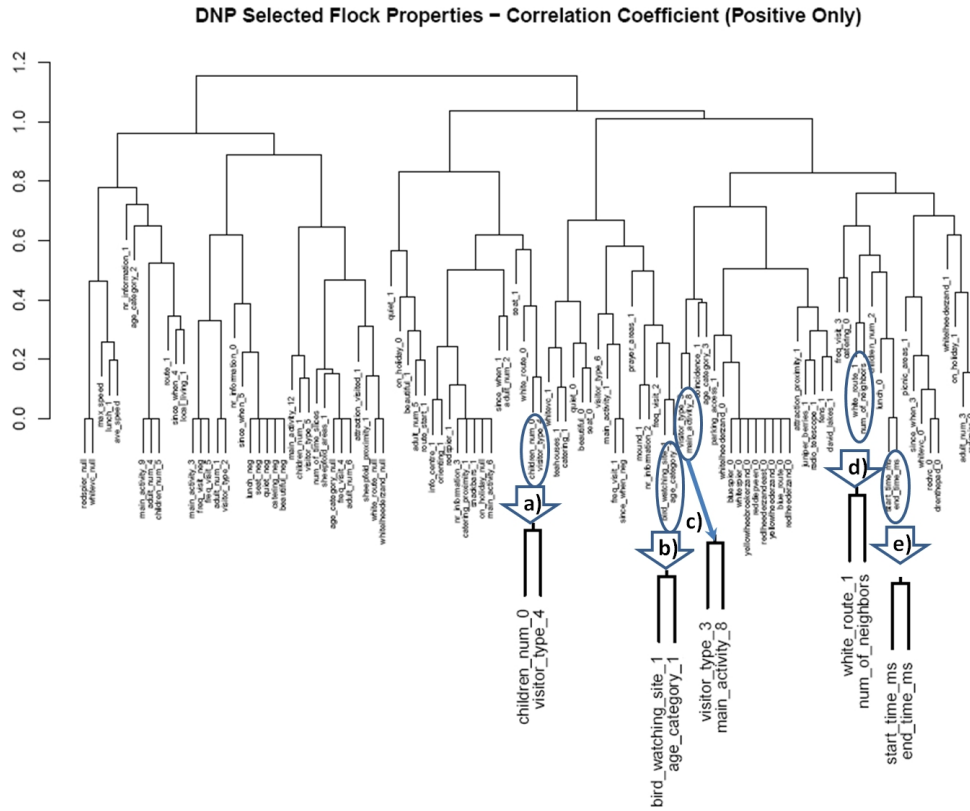


Figure 4.18: Sample relations among flock properties in semi-synthetic DNP.

Figure 4.19 presents the obtained relations among the discovered flock patterns. Recall that relations situated in the lower part of a dendrogram are more similar. As expected, *Flock_8*, *Flock_9* and *Flock_10* are most similar to each other and this is explained by the fact that they have 2 members in common. The trajectories belonging to *Flock_8* and *Flock_9* are shown in Figure 4.20, which confirms that the flocks are indeed similar based on their trajectories. *Flock_4* and *Flock_6*, on the other hand, have 1 member in common but they were not grouped together. Then, the rest of the flocks do not have any member in common. They were grouped together on the basis that they have certain attributes that are quite similar. For example, most of the members in *Flock_0* and *Flock_1* consist of adult couples, and were mainly involved in walking. Furthermore, most of their members have visited the juniper berries, the fens, the radio telescope, and the David lakes. These were inferred using the dendrogram of flock entries and the dendrogram of individual attributes. Figure 4.21 presents the trajectories belonging to *Flock_0* and *Flock_1*. This example demonstrates that flocks sharing the same semantic attributes may flock in different locations. In general, the results show that the approach may flag patterns as similar when they share the same members and hence, the same semantic attributes as well, or based on similar semantic attributes alone.

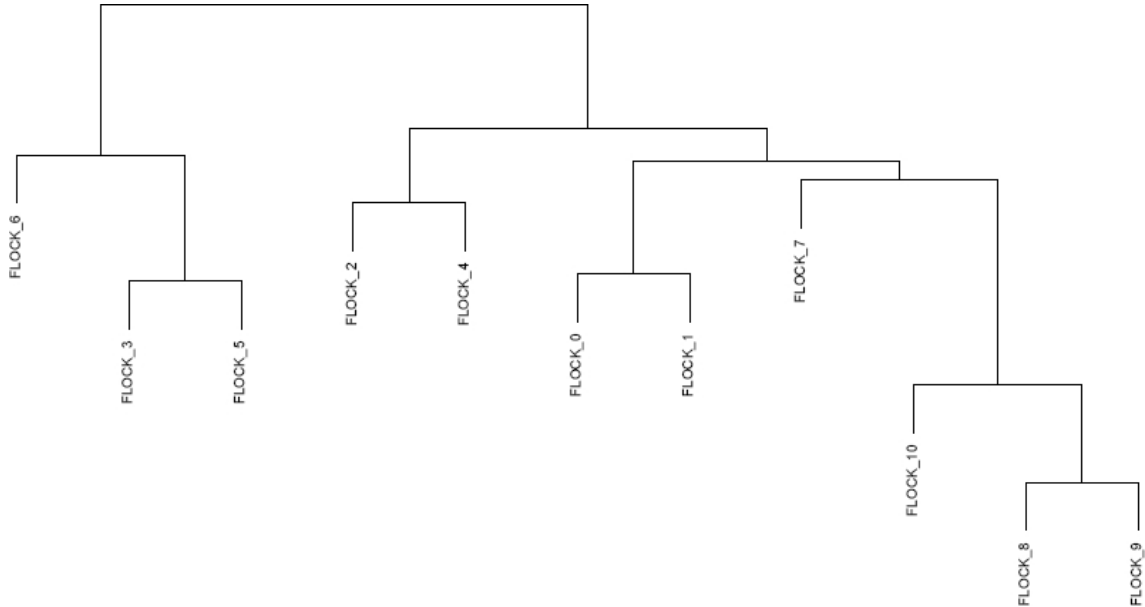


Figure 4.19: Relations among the flock patterns found in the analysis step.

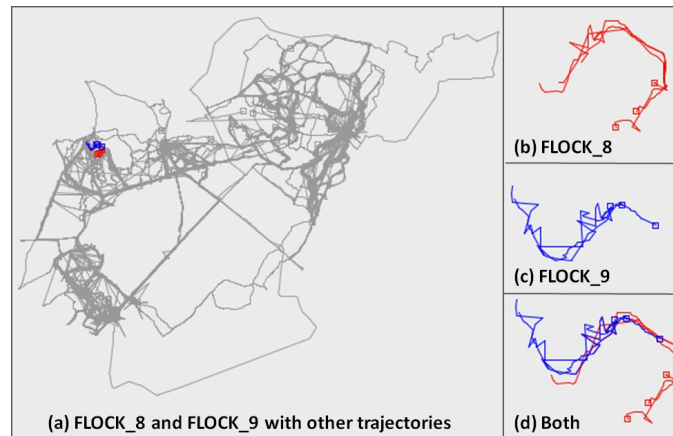


Figure 4.20: The trajectories of *FLOCK_8* and *FLOCK_9*.

Classification Results

Aside from performing hierarchical clustering, further analysis through the use of the J48 classification algorithm is employed in order to gain a deeper understanding of a subset of the discovered relations in the hierarchical clustering sub-step. While hierarchical clustering provides an overview of the existing relations found in the dataset, the classification step focuses on a subset of these relations and provides more details as to why these relations exist.

Analyzing Individual Attributes of Flock Members The J48 classification algorithm was run 11 times, one for each of the discovered flocks in order to obtain a decision tree for each flock. We will now describe some of the obtained decision trees, which sup-

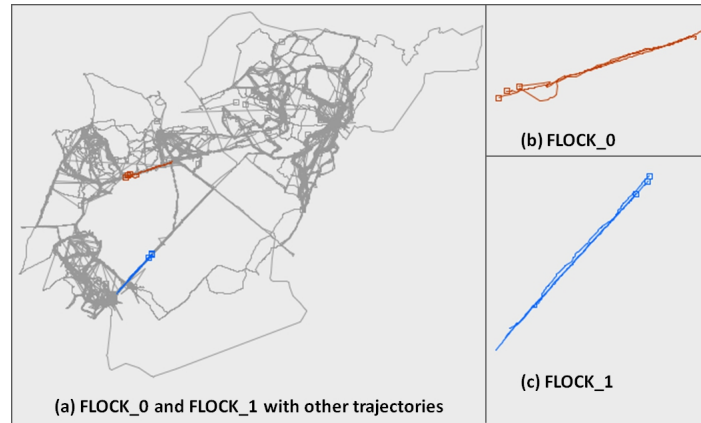


Figure 4.21: The trajectories of *FLOCK_0* and *FLOCK_1*.

port the previously described hierarchical clustering results, and the interpretations that can be inferred from them.

Figure 4.22 presents 2 decision trees, which were obtained when the target class was set to *Flock0* (left) and to *Flock2* (right). The decision tree on the left shows how *Flock0* is related to *radio_telescope* and other attributes. It is worth noting that the hierarchical clustering result has shown that there is an association between the flock and the radio telescope attraction. The details of this association is explained further with the obtained decision tree. It shows that members of *Flock0* have visited the radio telescope attraction and they were not interested in the parking accessibility in DNP. Additionally, each member consists of a couple who are either adults or elderly. Likewise, while hierarchical clustering results have shown that *Flock2* is associated with *mound* and *bird_watching_site*, more information can be obtained by performing classification analysis. The obtained decision tree on the right explains that members of *Flock2* have either visited the mounds or the bird watching sites.

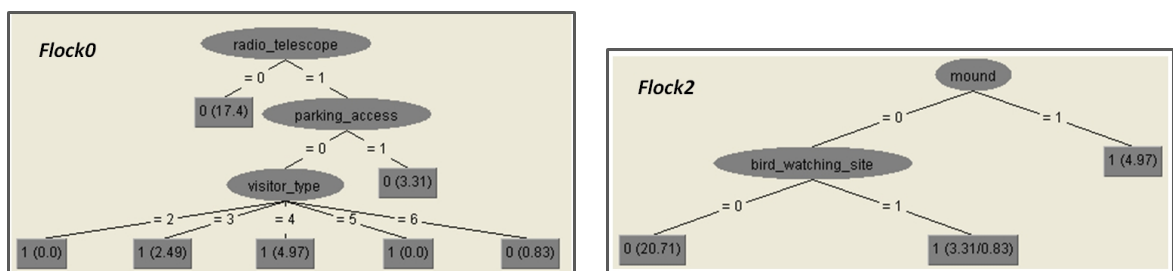


Figure 4.22: Decision tree obtained based on individual attributes of flock members when the target class is *Flock0* and *Flock2*, respectively.

Another interesting decision tree obtained using J48 is shown in Figure 4.23. The target class in this case is *Flock9*. Once again, this decision tree further expounds on the relations also discovered using hierarchical clustering analysis. It indicates that members of *Flock9* can be classified into 2 groups. Those that do not belong to *Flock8* should follow the White route while members of *Flock8* should have visited the park before. Note that *since_when* = -1 means that there is no data on when the visitor last visited the park.

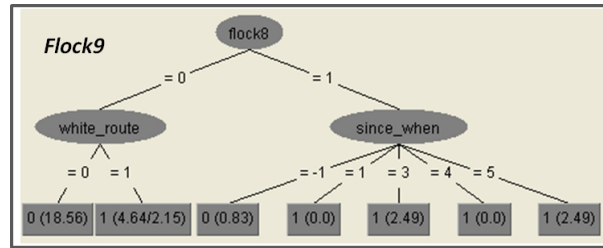


Figure 4.23: Decision tree obtained based on individual attributes of flock members when the target class is *Flock9*.

Analyzing Flock Attributes The J48 classification algorithm was also executed on flock attributes, setting the target class to the *main_activity* attributes. When the target class was set to *main_activity_1*, the decision tree shown in Figure 4.24 was obtained. *since_when_5* indicates the percentage of flock members who have visited the park more than 10 years before. Initially ignoring the branch corresponding to *since_when_5*=0.666667, one can interpret the decision tree as follows: if there are less members who have visited the park for more than 10 years ago, then there are likely more members whose main activity was walking. There are some cases though wherein this is not true as indicated by the branch that we initially ignored. In other words, the decision tree suggests that visitors who have visited the park more recently than 10 years ago have the tendency to have walking as their main park activity. This type of relations between main activities and other attributes can be derived from the other classification results as well.

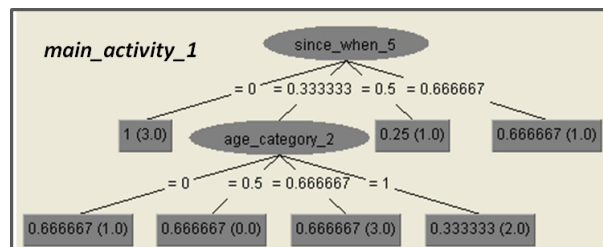


Figure 4.24: Decision tree obtained based on flock attributes when the target class is *main_activity_1*.

Summary of DNP Results

It is worth noticing that some promising results have been inferred despite of the limitations of the DNP dataset. Indeed, the dataset is quite small, and thus, the correlations found among the individual attributes, flock attributes and flock entries may or may not be conclusive. Besides being a small dataset, it is also quite sparse, containing mostly of either blank or 0 values (i.e., attribute is not satisfied) and this makes the distance measures less meaningful since the computed distance scores would be biased by the null values. However, the analysis performed on this data still shed light on certain interesting phenomena, which can be subjected for further verification by the domain experts.

The experiments demonstrate that further application of a classification algorithm provides a good support for explaining how relations found using the hierarchical clustering

step are correlated. Applying the described interpretation steps to trajectories of flock members allow the analyst to focus on relations that are only existent among visitors involved in flocking.

Combining the hierarchical clustering results for individual and flock attributes, the hierarchical clustering results for flock entries, and the classification results allow the analyst to infer interesting interpretation from the dataset.

4.4.2 Delft

For the application of the analysis phase to the Delft dataset, we focused on the 10 moving flocks that were obtained using the following set of parameter values: $min_points = 3$, $min_time_slices = 3$, $radius = 50m$, $synchronization_rate = 60s$. Each flock has 3 members each, and the members remain spatially close for 3-11 time instances (i.e., 2-10 minutes).

Correlation Computation and Hierarchical Clustering Results

Analyzing Individual Attributes of Flock Members Out of 296 trajectories in Delft, there are 24 trajectories that are members of some flock. We focused on the semantic attributes of these members for correlation computation and for hierarchical clustering.

Since 2 of the selected attributes (*postcodeb* and *occup*) contain string values, we only used SUC to compute the correlations among them. We have also replaced these 2 attributes with 2 other numeric attributes that are equivalent so as to be able to use Pearson's correlation coefficient aside from SUC. The hierarchical clustering results obtained using the different correlation measures were quite consistent. For this reason, we will only present sample relations obtained using the standard correlation coefficient, which allows the analyst to focus on positive correlations. Some of these interesting relations are presented in Figure 4.25. Relation **a**) shows that *flock2* is linked with *frequency* (i.e., refers to how often the pedestrian goes downtown) and they are closely related to *flock5* as well. This implies that members of both flocks tend to visit downtown more often. Relation **b**) shows that *flock1* is closely connected to *shopping*. The plot of the trajectories belonging to this flock is shown in Figure 4.26 with OpenStreetMap as background. It is important to point out that the flocking occurred in an area with cafés, restaurants, a church, a museum, and some shops. Referring back to Figure 4.25, relation **c**) links *flock6* with *wth_windy*. Finally, an interesting relation that involves 2 survey-based attributes is found in **d**). It indicates that *purpose* is linked to *gender*. These relations by themselves gives an idea of the composition of the flocks and the correlation among semantic attributes. Further details can be inferred by linking them with the other hierarchical clustering and the classification results.

Analyzing Flock Attributes For performing correlation computation and hierarchical clustering on flock attributes, the survey-based attributes used were derived from the selected individual attributes that includes the string attributes, *postcodeb* and *occup*. Since the flock attributes are all numerical, we used SUC and the 2 versions of Pearson's correlation coefficient for correlation computation. Once again, the dendograms obtained using the different measures were consistent and we only present some examples that were derived using Pearson's correlation coefficient for the purpose of focusing on positive

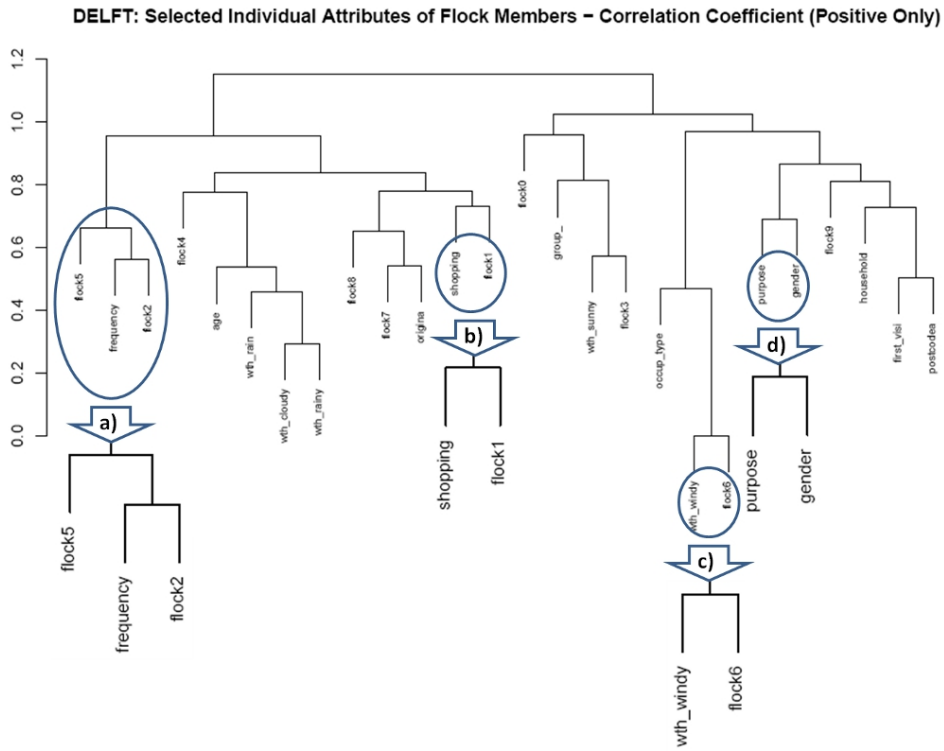


Figure 4.25: Sample relations among individual attributes of flock members in Delft.

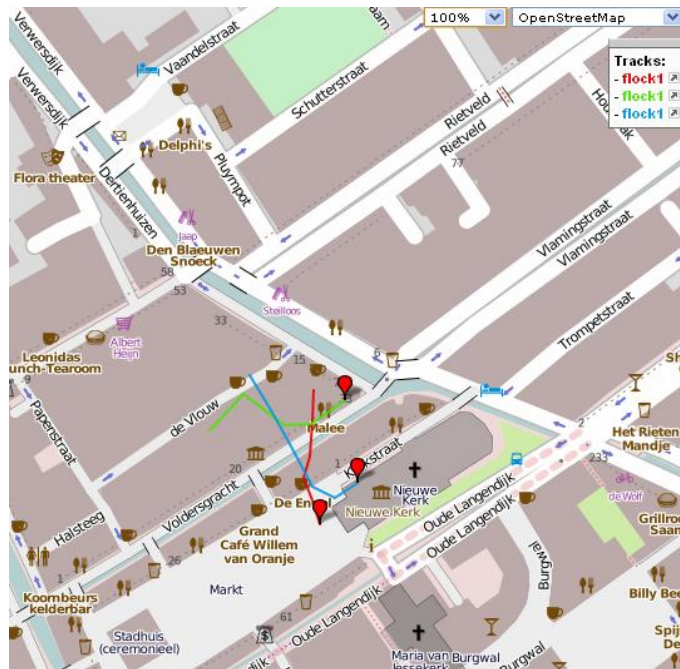


Figure 4.26: Trajectories belonging to *flock1* with OpenStreetMap as background.

correlations. Figure 4.27 presents some sample relations that were extracted. Sets **a)** and

b) show that categories *group_1* and *group_4* are linked with shopping and being 40 years old, respectively. Meanwhile, set c) connects gender with being retired. The first 2 sets can be explained further using classification analysis.

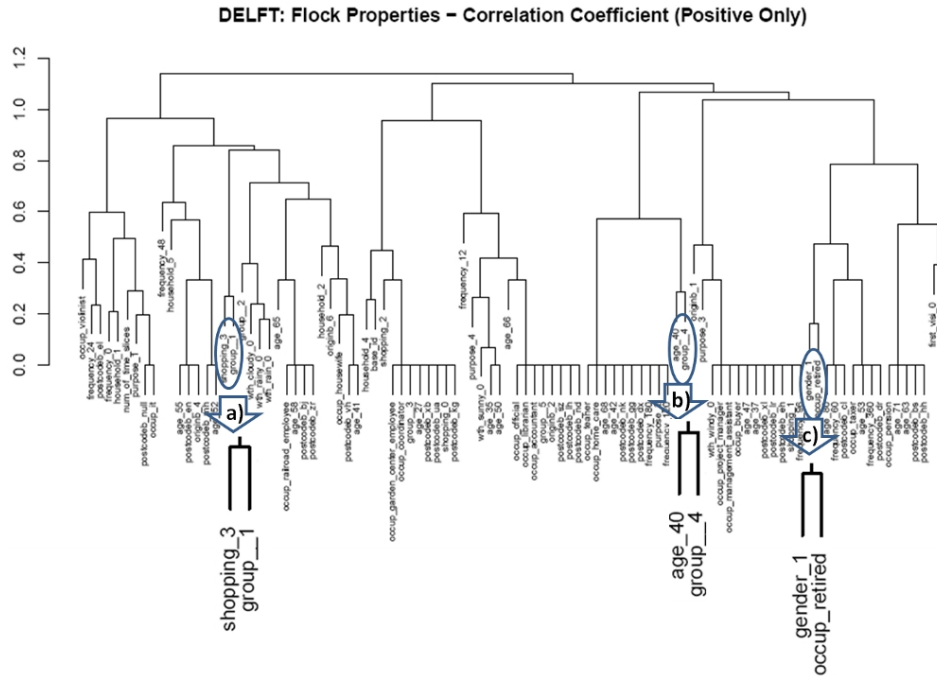


Figure 4.27: Sample relations among flock attributes in Delft.

Analyzing Flock Entries To understand the similarities among flock patterns themselves, we used the un-weighted and the weighted distance measures described in Section 3.3.1. Using both measures gave hierarchical clustering results that are almost similar. Figure 4.28 shows the relations among the discovered flocks using the weighted distance measures. The portion where this differs from the dendrogram obtained using the un-weighted measure is shown in Figure 4.29.

The flock pairs *Flock_2* and *Flock_5*, *Flock_7* and *Flock_8*, and *Flock_1* and *Flock_4* have 1 member in common while the rest of the flock groupings did not have any member in common. The reason why *Flock_3* and *Flock_9* were grouped together is because they have several semantic attributes with values that are almost similar, if not exactly similar. These include *purpose*, *frequency*, *originb*, *gender*, *age*, and the weather-related attributes.

Classification Results

As with the DNP experiment, J48 classification algorithm was also performed on the Delft dataset in order to gain a deeper understanding of the relations found using a combination of correlation computation and hierarchical clustering.

Analyzing Individual Attributes of Flock Members Classification analysis of individual attributes in the Delft dataset provided interesting relations, such as the example

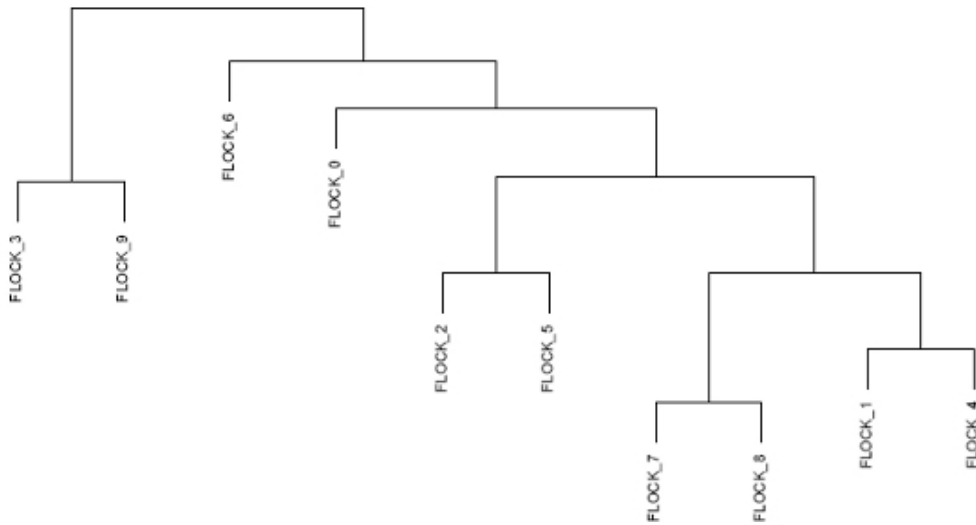


Figure 4.28: Relations among the flock patterns found in the analysis step.

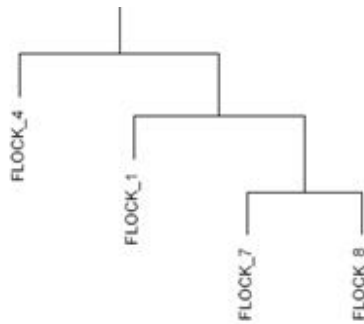


Figure 4.29: Subset of flock similarities in the un-weighted hierarchical clustering result.

provided in Figure 4.30. It describes an interesting relation between *Flock1* and *occup*. It is important to recall that using hierarchical clustering, *Flock1* was closely related to *shopping*. Combining these two results, we can infer that members of *Flock1* are mostly housewives and thus, are more involved in shopping activities. Without the combination of these 2 results, this interesting set of relations cannot be inferred by looking at each result alone.

Analyzing Flock Attributes Besides analyzing individual attributes, we have also performed classification analysis on the flock attributes. In particular, we have set the target class to the *group* attributes. This was executed 5 times since *group* has 5 possible values at individual level. The decision tree obtained by setting the target class to *group_1* is shown in Figure 4.31 while the tree for target class *group_4* is presented in Figure 4.32. These classification results expounds on the relations linked with *group_1* and *group_4*, which have been described earlier as sample relations obtained from the hierarchical clustering results. The decision tree for *group_1* indicates that this group is positively correlated with *shopping_3* and *gender_1*. On the other hand, *group_4* is associated with flocks wherein some members are 40 years of age and with some members

```

Flock1
occup = Taxer: 0 (0.8)
occup = IT: 0 (0.8)
occup = Housewife
| frequency = 0: 1 (0.0)
| frequency = 12: 1 (4.8)
| frequency = 24: 0 (1.6)
| frequency = 48: 1 (2.4)
| frequency = 60: 1 (0.0)
| frequency = 96: 1 (0.0)
| frequency = 120: 1 (0.0)
| frequency = 180: 1 (0.0)
| frequency = 360: 1 (0.0)
occup = Pension: 0 (0.8)
occup = Librarian: 0 (0.8)
occup = Retired: 0 (3.2)
occup = Home_care: 0 (0.8)
occup = Coordinator: 0 (0.8)
occup = Buyer: 0 (0.8)
occup = Teacher: 0 (0.8)
occup = Official: 0 (0.8)
occup = Accountant: 0 (0.8)
occup = Railroad_employee: 0 (0.8)
occup = Project_manager: 0 (0.8)
occup = Management_assistant: 0 (0.8)
occup = Violinist: 0 (0.8)
occup = Garden_center_employee: 0 (0.8)

```

Figure 4.30: Decision tree obtained based on individual attributes of flock members when the target class is *Flock1*.

coming from *origin_6*. For flocks that do not have any members who are 40 years old, they should not be linked with *shopping_2* in order to be categorized under *group_4*.



Figure 4.31: Decision tree obtained based on flock attributes when the target class is *group_1*.

Summary of Delft Results

Interesting relations were found by performing the analysis phase on the Delft dataset, such as the relation between *Flock1* and moving entities who are housewives and thus, are involved in shopping activities. As with DNP, the combination of hierarchical clustering results on the individual and flock attributes and on the flock entries with the classification results allowed the analyst to deduce interpretations that could not have been inferred if each results were analyzed alone.

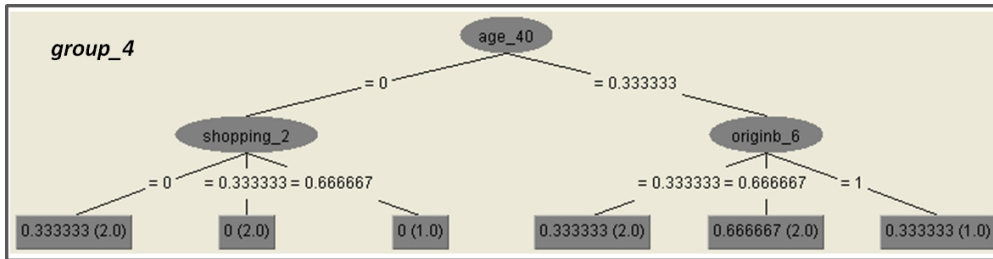


Figure 4.32: Decision tree obtained based on flock attributes when the target class is *group_4*.

4.5 Overall Summary

This section summarizes the results obtained using the pattern interpretation framework.

By applying the pattern discovery phase of the framework, we were able to find a number of moving flocks from the DNP, Fontainebleau and Delft datasets. Our initial experiments included stationary flocks in the results and we found that the number of such flocks can be too much for further analysis (recall Figures 4.3, 4.6, and 4.9). Moreover, many of these flocks are not real flocks in the sense that they may be slightly moving in different directions and stopping in a certain location. This type of flocks fall under another class of patterns called meet, which was described earlier in Section 2.3. Compared with existing flock algorithms, our algorithm provides the advantage of allowing the user to focus on patterns that can lead to understanding the aggregate movement behavior of entities over a sequence of locations.

The discovered moving flock patterns by themselves do not provide much information about the behavior of the flocking entities. The results only provide information about the geographical coordinates, and the time of flocking. In order to exploit these patterns for the purpose of understanding movement behavior, the semantic annotation and the pattern analysis phases of the framework must be applied. By doing so, we were able to find some interesting interpretations from both the DNP and the Delft datasets.

In the DNP dataset, we were able to identify common properties shared by individuals who go to the same place at similar times of the day. For example, members of *flock0* are attracted to the park for no specific reasons and yet they have visited several attractions such as *radio_telescope*, *david_lakes*, *juniper_berries*, and *fens*. They were in the same location around lunch time at different days. We were also able to relate properties such as visitor types, age category, main activities and attraction visited. For instance, flocking members whose age ranges from 30-59 tend to visit the bird watching site. Furthermore, the analysis of the flock entries allowed the identification of similar or related flocks. An example of such flocks are *Flock_8*, *Flock_9* and *Flock_10*, which are similar due to the presence of common members. *Flock_0* and *Flock_1* are also similar but in the semantic sense, having many members that are adult couples and were mainly involved in walking.

Likewise, interesting interpretations were also obtained from the Delft dataset. For example, a common property shared by members of *flock1* is their interest in shopping and most members of this flock are housewives. This result is validated by the fact that the flocking occurred in an area with different types of shops and services such as music shops, shoe store, supermarket, cafés and restaurants. Another interesting relation we

found is between the purpose and the gender of the flocking pedestrians, which is sound since male and female pedestrians tend to have different purposes in moving.

The interpretations found in both datasets were made possible with the integration of semantic data to the discovered patterns and the analysis of the semantically enriched patterns. The performed experiments demonstrate that interesting interpretations can be inferred through the utilization of the proposed pattern interpretation framework.

Chapter 5

Conclusions

This chapter provides a discussion of the contributions that have been accomplished through the the proposed framework and through its individual phases. We close with a discussion of the possible directions for extending the proposed framework.

5.1 Contributions

In this thesis, we emphasize the importance of both the data mining and the interpretation phases of the KDD process by proposing a framework that covers both phases and demonstrating the type of interpretations that can be inferred using this framework on real world datasets.

The succeeding paragraphs emphasize the main contributions of our work.

New Notion of Moving Flocks We focused on a specific type of movement pattern called moving flock patterns, which is a novel concept we have initially introduced in [89]. While existing definitions of flock patterns exist, it was only in our definition of moving flock patterns that the moving constraint was introduced. This constraint restricts moving flock patterns to refer to a group of objects that move together from one location to another over a maximal time period, in contrast to simply staying together in one location. Therefore, we differentiate between moving and stationary flocks since we perceive them as two different types of patterns. Consequently, we perceive them as having different semantics.

The Framework Another novelty in this work is the proposal of a pattern interpretation framework that includes both the discovery of patterns and their analysis. This provides a more complete picture of the KDD process for flock patterns compared to existing works by exploiting semantic attributes that are explicitly included in the dataset. As a result, the use of the framework allows the deduction of meaningful interpretations that are useful in understanding movement behaviors.

Moreover, the framework initiates a first step towards interpretation of movement patterns. While there are few existing works that attempt to address this issue as described in Section 2.6, the state of the art in this field is still at the incubation stage. Recall that existing works can be categorized into two groups, namely, visual analytics tools and ontology-based systems. While visual analytics tools have helped in moving a step closer

to understanding patterns, such tools can be improved further by considering other types of semantic data, such as specific characteristics of the moving entities, aside from mainly relying on geographical data for semantics. Meanwhile, most ontology-based systems that aim to interpret movement patterns mainly classify the extracted patterns based on the semantics incorporated to the data through ontology. For example, a pattern can be classified as tourist-related or work-related using such systems but support for deeper interpretations can further ease the task of the analyst. On the other hand, our framework supports deeper interpretation by exploiting thematic attributes included in the input data and by utilizing data mining techniques. This allows the deduction of interactions among moving entities and possible reasons for the existence of the pattern.

Instantiation-specific Contributions Aside from introducing a conceptual framework for pattern interpretation, we have also demonstrated how the framework can be instantiated to interpret specific patterns, particularly moving flock patterns. The next paragraphs describe the specific accomplishments, which are obtained through this instantiation and are discussed according to each phase of the framework.

Pattern Discovery Phase: Aside from formally defining the concept of moving flocks for the pattern discovery phase, we have also developed and implemented an algorithm that extracts such patterns from datasets consisting of (x, y, t) observations. The algorithm considers spatio-temporal coherence among candidate flock members by checking their spatial closeness as defined by a circular region of predefined radius with respect to a base object at each time instance. Afterwards, spatially close objects whose spatial coherence persists over a period of consecutive time instances are merged to complete the check for spatio-temporal coherence. The distinguishing feature of this algorithm compared to existing flock algorithms is its pruning step, which involves the elimination of redundant and stationary patterns.

The algorithm was validated through the visualization of flock trajectories and through an application of the null hypothesis principle in order to confirm that the obtained flock results are inherent in the dataset. In applying this principle, two techniques for randomization techniques for movement dataset were described. These include randomization using Markov chain and randomization based on uncertainties in (x, y) data. Moreover, the algorithm was shown to be robust with respect to entity ordering in the input dataset. Though the results may vary at times, the variations are few and mostly minor.

Furthermore, we recognize that the selection of the algorithm's parameters is crucial in obtaining meaningful flocks. Thus, we have described a set of guidelines for selecting the parameters and we have extended a technique used in DBSCAN in order to specifically suggest a good radius value to the end user.

Semantic Annotation Phase: As for the semantic annotation phase, we proposed a guideline for selecting attributes to be used for semantic annotation by referring to the work in [92], which do not only cover flocks but collectives in general. Moreover, we propose two levels of semantic annotation, one at individual level and another at flock pattern level. For the individual level, it would be ideal to only consider the properties of flock members and ignore those that do not belong to any flock. This is advantageous since only a subset of entities have to be annotated and analyzed, which allows the analyst to

focus on relations that exist among individual attributes when flocking occurs. However, in the case that the dataset is small, statistically significant results may not be inferred by focusing only on such a small subset of individuals. Thus, an alternative is to consider individual properties of all individuals, whether they belong to a flock or not, before analyzing the relations that occur among them. However, doing this extracts relations that may not exist for flocking entities.

These specific contributions (i.e., the guideline for semantic attribute selection and the two levels of semantic annotation) of the semantic annotation phase addresses two issues that were posed as questions in Section 1.2. Recall that these questions are the following:

1. At which level should the movement data and patterns be annotated?
2. Which of the available semantic attributes should be incorporated into the movement data and patterns?

Pattern Analysis Phase: Lastly, for the pattern analysis phase, we propose the combination of a hierarchical clustering and a decision tree induction classification algorithm in order to interpret the patterns found in the pattern discovery phase. While these data mining tasks are typically applied to the entries found in the dataset, we chose to apply them to individual properties, to flock properties, and to flock patterns themselves. Doing so can aid the analyst in understanding extracted flock patterns. Clustering allows the analyst to have an overview of the different relations that exist among the considered attributes and among the flocks, while classification allows the analyst to focus on the more interesting relations and extract more details as to why the correlation exists among them. We chose to use hierarchical clustering over other types of clustering algorithms since it allows the analyst to pinpoint stronger over weaker relations and hence, having the capability to determine which relations are more important. Hierarchical clustering results also provide information about the relations among different clusters. On the other hand, we selected a decision tree induction algorithm since this type of algorithm provides classification results that are intuitive and easy to understand.

Since hierarchical clustering requires a distance matrix of the compared entries, we used SUC and Pearson's correlation coefficient to compute the similarity among attributes and extract the distance matrix from this. The use of a combination of different correlation measures allow the analyst to determine which relations are more interesting by finding those that are consistent among all measures. As for computing the similarity among flocks, we have used a straightforward computation by averaging the difference among corresponding flock attribute values. We have also described a taxonomy-based approach that matches semantically corresponding attributes, besides those that exactly correspond, for computing flock similarity.

Note that the instantiation of the three phases of the framework to moving flock interpretation addresses the following questions posed as challenges in Section 1.2:

1. How can the semantically-enriched movement data and pattern be transformed into meaningful patterns?
2. What type of interpretations can be inferred using this technique?

The details provided for each phase of the framework answers the first question. To address the second question, it was mentioned that application of the instantiated framework allows the deduction of interactions among moving flock members, which can lead to understanding flocking behaviors. It is important to note, however, that the interpretations deduced do not have full certainty. We have currently addressed the certainty issue by checking the consistency of the clustering and classification results. That is, if a relation is obtained from the clustering and the clustering regardless of whether SUC or Pearson's coefficient was utilized, then this relation has a higher certainty of being meaningful compared to a relation obtained from either clustering or classification result alone.

Application to Real-World Datasets: To assess the feasibility and the usefulness of the framework, we have developed a set of tools to support its application and it was tested on two pedestrian datasets, which are the DNP and the Delft dataset. One of the interesting type of interaction inferred from the DNP dataset is the tendency of the flocks to follow the White route, which is the most popular path in the park according to the domain expert. In the Delft dataset, an interesting relation obtained is the connection between a pedestrian's gender and his/her purpose in moving. In addition to these datasets, the moving flock algorithm was also tested on the Fontainebleau dataset.

Limitations A main limitation of the framework, however, is its scalability. It is good for targeted analysis wherein the user is interested in a specific and small set of patterns. However, the semantic annotation and the pattern analysis phases can become cumbersome if there is a large number of semantic attributes and/or a large number of patterns to process. More sophisticated techniques that addresses this issue should be designed and developed.

Another limitation lies in the interpretation results that can be obtained using the instantiated framework. Though it allows the analyst to infer interactions among moving entities, there is currently no quantitative measure that assesses the interpretations' certainty. Aside from relying on the domain expert or the consistency of the obtained results, it would be ideal to have a measure that evaluates the interpretation result.

5.2 Future Works

Based on the known limitations of the proposed framework, this section provides a discussion of possible extensions to improve it.

Though the proposed framework was only completely tested on pedestrian datasets and on moving flock patterns, it is applicable to other datasets as well and can be applied in discovering and interpreting other pattern types. Unfortunately, there are currently only a small number of datasets that explicitly include semantic attributes. Moreover, semantic data is also prone to human error. Analysis of such data, assessing and improving their accuracy is another aspect of the semantic annotation phase that should be further investigated. We are currently testing the flock discovery algorithm on a vehicles dataset. The semantics of the discovered flocks would change in this case. In the pedestrian datasets, flocks are interpreted as groups following certain paths/routes together. This type of behavior is less common among vehicles and flocking among vehicles more commonly occurs due to traffic jams. The appropriate set of parameters for vehicles is

also different compared to those for pedestrians.

As mentioned previously, it would also be interesting to apply the framework to consider other pattern types. We have associated stationary flocks with meet, convergence and encounter patterns since these patterns describe a group of moving entities headed towards the same location. For this reason, these patterns are closely related to flocks and can be interpreted using the instantiation of the framework for moving flocks. Aside from applying to flock-related patterns, it would also be interesting to apply it to other patterns, such as the T-pattern. Recall that a T-pattern represents the typical collective movement between regions of interest. Since the identity of individual members involved in this pattern is perceived as less important and the emphasis is on the regions of interest, the semantic annotation and the pattern interpretation phases should focus on geographical information in order to understand the meaning implicit in such types of patterns. This can lead to interpretations on how the geographical environment affects the movement behavior implicit in the data.

The dependence of the framework's success on the explicit availability of semantic attributes of the dataset can be seen as a limitation. In cases where such semantics are not explicitly available, it would be ideal to have a tool that automatically integrates semantic attributes into the dataset based on other sources of context information. Semi-automatic annotation tools can also be useful in this integration task.

Another limitation of the framework is its scalability in terms of interpreting large numbers of patterns with large number of semantic properties, as mentioned in the previous section. The following directions can be pursued to address this issue. First, the implementation of the flock discovery algorithm can be improved by the using faster search algorithm and indices in order to make the computation of the spatial neighborhood more efficient. Second, the semantic annotation phase can be semi-automated through the development and use of suitable annotation tools, and through the introduction of ontologies. More specifically, refinements can be introduced in Wood and Galton's set of criteria [92] in order to achieve the right level of distinguishing criteria for particularly classifying moving flocks. Having such a refined set of criteria can filter a large number of unnecessary attributes. Third, incorporating a refined taxonomy for moving flocks, or at least flocks in general, to the pattern interpretation phase may ease the task of inferring meanings from patterns and also improve the quality of deduced interpretations. In this case, we foresee this future work to be quite related to ontology-based interpretation systems. Lastly, an integrated tool that supports the entire framework can ease the process. This tool should support progressive analysis of the discovered flocks by allowing users to focus on specific flock types based on an interesting set of criteria such as flocks occurring in certain attractions, flocks occurring on certain time durations, and others.

The moving flock algorithm can be improved further by considering density-based algorithms that do not restrict the shape of an entity's spatial neighbors. Currently, the algorithm uses a disk to approximate the spatial neighbors of an entity. As a consequence spatial neighborhoods are restricted to a circular shape, which in turn, may result in the extraction of flocks that include some members as noise (i.e., these are not really members of the flock but are included as members due to the shape of the disk). Using a flexible shape for the spatial neighborhood can help in obtaining flocks that are less susceptible to noises. However, algorithms for finding neighbors with such shapes are less efficient. Moreover, a possible problem with using unrestricted shapes is that a single shape may capture more than one flocking behavior, making the analysis of discovered patterns more

complex.

In order to address the issue of assessing the obtained interpretation results, there is a need for a domain expert or some form of stored knowledge in order to perform supervised validation of the results.

References

- [1] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic. Discovering Clusters in Motion Time-Series Data. In *Proceedings of the 2003 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)*, pages 375–381, Los Alamitos, CA, 2003. IEEE.
- [2] L. Alvares, V. Bogorny, B. Kuijpers, J. de Macedo, B. Moelans, and A. Vaisman. A Model for Enriching Trajectories with Semantic Geographical Information. In *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems (GIS'07)*, pages 1–8, New York, 2007. ACM.
- [3] L. Alvares, V. Bogorny, B. Kuijpers, B. Moelans, J. de Macedo, and A. Palma. Towards Semantic Trajectory Knowledge Discovery, 2007.
- [4] M. Andersson, J. Gudmundsson, P. Laube, and T. Wolle. Reporting leaders and followers among trajectories of moving point objects. *GeoInformatica*, 12(4):497–528, 2008.
- [5] G. Andrienko, N. Andrienko, and S. Wrobel. Visual Analytics Tools for Analysis of Movement Data. *SIGKDD Explor. Newsl.*, 9(2):38–46, 2007.
- [6] N. Andrienko and G. Andrienko. Designing Visual Analytics Methods for Massive Collections of Movement Data. *Cartographica*, 42(2):117–138, 2007.
- [7] N. Andrienko, G. Andrienko, N. Pelekis, and S. Spaccapietra. *Basic Concepts of Movement Data*. In F. Gianotti and D. Pedreschi, (Eds.), *Mobility, Data Mining, and Privacy: Geographic Knowledge Discovery*, pages 15–38. Springer–Verlag, Berlin Heidelberg, 2008.
- [8] ArcGIS: A Complete Integrated System. <http://www.esri.com/software/arcgis/index.html>.
- [9] Australian Bureau of Statistics. <http://www.censusdata.abs.gov.au>.
- [10] K. Axhausen, S. Schönfelder, J. Wolf, M. Oliveira, and U. Samaga. 80 Weeks of GPS-Traces: Approaches to Enriching the Trip Information, 2003.
- [11] M. Baglioni, J. de Macêdo, C. Renso, R. Trasarti, and M. Wachowicz. Towards Semantic Interpretation of Movement Behavior. In W. Cartwright, G. Gartner, L. Meng, and M. Peterson, editors, *Advances in GIScience*, Lecture Notes in Geoinformation and Cartography, pages 271–288. Springer Berlin Heidelberg, 2009.

- [12] M. Baglioni, J. de Macêdo, C. Renso, and M. Wachowicz. An Ontology-Based Approach for the Semantic Modelling and Reasoning on Trajectories. In I. Song, M. Piattini, Y. Chen, S. Hartmann, F. Grandi, J. Trujillo, A. Opdahl, F. Ferri, P. Grifoni, M. Caschera, C. Rolland, C. Woo, C. Salinesi, E. Zimányi, C. Claramunt, F. Frasin-car, G. Houben, and P. Thiran, editors, *Advances in Conceptual Modeling Challenges and Opportunities*, volume 5232 of *Lecture Notes in Computer Science*, pages 344–353. Springer Berlin / Heidelberg, 2008.
- [13] Basic formal ontology (bfo). <http://www.ifomis.org/bfo>.
- [14] M. Batty, J. Desyllas, and E. Duxbury. The Discrete Dynamics of Small-scale Spatial Events: Agent-based Models of Mobility in Carnivals and Street Parades. *International Journal of Geographical Information Science*, 17(7):673–697, 2003.
- [15] M. Benkert, J. Gudmundsson, F. Hbner, and T. Wolle. Reporting Flock Patterns. In *Proceedings of the 14th European Symposium on Algorithms (ESA '06)*, pages 660–671. Springer, 2006.
- [16] L. Bian. A Conceptual Framework for an Individual-based Spatially Explicit Epidemiological Model. *Environment and Planning B*, 31(3):381–396, 2004.
- [17] E. Bottazzi, C. Catenacci, A. Gangemi, and J. Lehmann. From Collective Intentionality to Intentional Collectives: an Ontological Perspective. *Cognitive Systems Research*, 7(2-3):192–208, June 2006.
- [18] R. Buliung and P. Kanaroglou. An Exploratory Spatial Data Analysis (ESDA) Toolkit for the Analysis of Activity/Travel Data. In A. Laganá, M. Gavrilova, V. Kumar, Y. Mun, C. Tan, and O. Gervasi, editors, *Computational Science and Its Applications (ICCSA '04)*, volume 3044 of *Lecture Notes in Computer Science*, pages 1016–1025. Springer Berlin / Heidelberg, 2004.
- [19] Census – Wikipedia. <http://en.wikipedia.org/wiki/Census>.
- [20] D. Chudova, S. Gaffney, E. Mjolsness, and P. Smyth. Translation-Invariant Mixture Models for Curve Clustering. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, 2003. ACM.
- [21] V. Colizza, A. Barrat, M. Barthelemy, A. Valleron, and A. Vespignani. Modeling the Worldwide Spread of Pandemic Influenza: Baseline Case and Containment Interventions. *PLoS Medicine*, 4(1):95–110, 2007.
- [22] J. de Macedo, C. Vangenot, W. Othman, N. Pelekis, E. Frentzos, B. Kuijpers, I. Ntoutsis, S. Spaccapietra, and Y. Theodoridis. *Trajectory Data Models*. In F. Giannotti and D. Pedreschi, (Eds.), *Mobility, Data Mining, and Privacy: Geographic Knowledge Discovery*, pages 123–150. Springer-Verlag, Berlin Heidelberg, 2008.
- [23] Delft – Wikipedia. <http://en.wikipedia.org/wiki/Delft>.
- [24] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. Tomlin, and J. Zien. SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation. In *Proceedings of the 12th*

- International Conference on World Wide Web, WWW '03*, pages 178–186, New York, NY, USA, 2003. ACM.
- [25] S. Dodge, R. Weibel, and A. Lautenschütz. Towards a Taxonomy of Movement Patterns. *Information Visualization*, 7:240–252, June 2008.
- [26] E. Galea (Ed.). *Pedestrian and Evacuation Dynamics*. CMS Press, Greenwich, UK, 2003.
- [27] M. Erdmann, A. Maedche, H. Schnurr, and S. Staab. From Manual to Semi-automatic Semantic Annotation About Ontology-based Text Annotation Tools. In *P. Buitelaar & K. Hasida (eds). Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content*, Luxembourg, August 2000.
- [28] M. Ester, H. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In E. Simoudis, J. Han, and U. Fayyad, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.
- [29] U. Fayyad, G. Piatesky-Shapiro, and P. Smyth. The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Communications of the ACM*, 39:27–34, November 1996.
- [30] Fontainebleau. <http://www.uk.fontainebleau-tourisme.com/pays-fontainebleau/town-and-history/fontainebleau.asp>.
- [31] S. Gaffney and P. Smyth. Trajectory Clustering with Mixtures of Regression Models. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 63–72, New York, 1999. ACM.
- [32] Geodatabase. <http://www.esri.com/software/arcgis/geodatabase/index.html>.
- [33] Geography home page - about.com. <http://geography.about.com/>.
- [34] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory Pattern Mining. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07)*, pages 330–339, New York, 2007. ACM.
- [35] F. Gianotti and D. Pedreschi (Eds.). *Mobility, Data Mining and Privacy: Geographic Knowledge Discovery*. Springer-Verlag, Berlin Heidelberg, 2008.
- [36] B. Guc, M. May, Y. Saygin, and C. Körner. Semantic Annotation of GPS Trajectories. In *11th AGILE International Conference on Geographic Information Science*, 2008.
- [37] J. Gudmundsson and M. van Kreveld. Computing Longest Duration Flocks in Trajectory Data. In *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems (GIS'06)*, pages 35–42, New York, NY, USA, 2006. ACM.
- [38] J. Gudmundsson, M. van Kreveld, and B. Speckmann. Efficient Detection of Motion Patterns in Spatio-Temporal Data Sets. In *Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems (GIS'04)*, pages 250–257, New York, NY, USA, 2004. ACM.

- [39] M. Hall. Correlation-based Feature Selection for Machine Learning. In *Ph.D. Thesis, The University of Waikato, Hamilton, New Zealand*, 1999.
- [40] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA Data Mining Software: an Update. *SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
- [41] David J. Hand, Padhraic Smyth, and Heikki Mannila. *Principles of Data Mining*. MIT Press, Cambridge, MA, USA, 2001.
- [42] D. Helbing, L. Buzna, A. Johansson, and T. Werner. Self-organized Pedestrian Crowd Dynamics: Experiments, Simulations, and Design Solutions. *Transportation Science*, 39(1):1–24, 2005.
- [43] S. Hwang, Y. Liu, J. Chiu, and E. Lim. Mining Mobile Group Patterns: A Trajectory-Based Approach. In *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'05)*, pages 713–718, Berlin Heidelberg, 2005. Springer.
- [44] H. Yong J. Kang. Mining Trajectory Patterns by Incorporating Temporal Properties. In *Proceedings of the 1st International Conference on Emerging Databases*, 2009.
- [45] J48. <http://www.opentox.org/dev/documentation/components/j48>.
- [46] H. Jeung, M. Yiu, X. Zhou, C. Jensen, and H. Shen. Discovery of Convoys in Trajectory Databases. *Proceedings of the VLDB Endowment*, 1(1):1068–1080, 2008.
- [47] P. Kalnis, N. Mamoulis, and S. Bakiras. On Discovering Moving Clusters in Spatio-Temporal Data. In *Proceedings of the 9th International Symposium on Spatial and Temporal Databases (SSTD'05)*, pages 364–381, Berlin Heidelberg, 2005. Springer.
- [48] J. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting Places from Traces of Locations. In *Proceedings of the 2nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH'04)*, pages 110–118, New York, NY, USA, 2004. ACM.
- [49] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff. Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2(1):49 – 79, 2004.
- [50] S. Kisilevich, F. Mansmann, M. Nanni, and S. Rinzivillo. Spatio-temporal Clustering. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 855–874. Springer US, 2010.
- [51] P. Laube, M. van Kreveld, and S. Imfeld. *Finding Remo - Detecting Relative Motion Pattern in Geospatial Lifelines*. In P. Fisher, (Ed.), *Developments in Spatial Data Handling.*, pages 201–215. Springer-Verlag, Berlin Heidelberg, 2005.
- [52] J. Lee, J. Han, and K. Whang. Trajectory Clustering: A Partition-and-Group Framework. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data (SIGMOD'07)*, pages 593–604, New York, 2007. ACM.

- [53] Y. Li, J. Han, and J. Yang. Clustering Moving Objects. In *Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining (KDD'04)*, pages 617–622, New York, 2004. ACM.
- [54] D. Lin. An Information-Theoretic Definition of Similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML'98)*, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
- [55] J. MacQueen. Some Methods for Classification and Analysis of MultiVariate Observations. In L. Le Cam and J. Neyman, editors, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
- [56] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. WonderWeb deliverable D18 ontology library (final). Technical report, IST Project 2001-33052 WonderWeb: Ontology Infrastructure for the Semantic Web, 2003.
- [57] R. May and A. McLean. *Theoretical Ecology: Principles and Applications*. Oxford University Press, New York, 2007.
- [58] Q. Mei, D. Xin, H. Cheng, J. Han, and C. Zhai. Generating Semantic Annotations for Frequent Patterns with Context Analysis. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*, pages 337–346, New York, NY, USA, 2006. ACM.
- [59] Merriam-Webster OnLine. <http://www.merriam-webster.com/dictionary>.
- [60] H. Miller and J. Han (Eds.). *Geographic Data Mining and Knowledge Discovery*. Taylor & Francis, Bristol, PA, USA, 2001.
- [61] D. Mountain and J. Raper. Modelling Human Spatio-Temporal Behaviour: a Challenge for Location based Services. In *Proceedings of the 6th International Conference on GeoComputation*, 2001.
- [62] Movement Patterns. <http://movementpatterns.pbworks.com/w/page/21692526/FrontPage>.
- [63] Museum planner blog survey results. <http://museumplanner.org/museum-planner-blog-survey-results/>.
- [64] M. Nanni and D. Pedreschi. Time-Focused Clustering of Trajectories of Moving Objects. *Journal of Intelligent Information Systems*, 27(3):267–289, 2006.
- [65] Object Management Group – UML. <http://www.uml.org/>.
- [66] R. Ong, M. Wachowicz, M. Nanni, and C. Renso. From Pattern Discovery to Pattern Interpretation in Movement Data. In *IEEE ICDM Workshop Proceedings*, pages 527–534, 2010.
- [67] Oracle Database 11g. <http://www.oracle.com/us/products/database/index.html>.

- [68] R. Ortale, E. Ritacco, N. Pelekis, R. Trasarti, G. Costa, F. Giannotti, G. Manco, C. Renso, and Y. Theodoridis. The DAEDALUS Framework: Progressive Querying and Mining of Movement Data. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS'08)*, pages 1–4, New York, NY, USA, 2008. ACM.
- [69] A. Palma, V. Bogorny, B. Kuijpers, and L. Alvares. A Clustering-based Approach for Discovering Interesting Places in Trajectories. In *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC'08)*, pages 863–868, New York, 2008. ACM.
- [70] Patterns of Movement Wiki. <http://movementpatterns.pbworks.com/w/page/21692527/Patterns-of-Movement>.
- [71] Pearson Product-Moment Correlation Coefficient – Wikipedia. http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient.
- [72] N. Pelekis, I. Kopanakis, G. Marketos, I. Ntoutsis, G. Andrienko, and Y. Theodoridis. Similarity search in trajectory databases. In *Proceedings of the 14th International Symposium on Temporal Representation and Reasoning (TIME'07)*, pages 129–140, Washington, DC, USA, 2007. IEEE Computer Society.
- [73] N. Pelekis and Y. Theodoridis. Boosting Location-based Services with a Moving Object Database Engine. In *Proceedings of the 5th ACM International Workshop on Data Engineering for Wireless and Mobile access (MobiDE'06)*, pages 3–10, New York, NY, USA, 2006. ACM.
- [74] Protégé. <http://protege.stanford.edu/>.
- [75] J. Quinlan. Learning With Continuous Classes. In *5th Australian Joint Conference on Artificial Intelligence*, pages 343–348. World Scientific, 1992.
- [76] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, USA, 1993.
- [77] The R Project for Statistical Computing. <http://www.r-project.org/>.
- [78] L. Reeve and H. Han. Survey of Semantic Annotation Platforms. In *Proceedings of the 2005 ACM Symposium on Applied Computing, SAC '05*, pages 1634–1638, New York, NY, USA, 2005. ACM.
- [79] IEEE - SADM 2010. <http://sadm2010.isti.cnr.it/>.
- [80] M. Schreckenberg and S. Sharma (Eds.). *Pedestrian and Evacuation Dynamics*. Springer-Verlag, 2002.
- [81] S. Spaccapietra, C. Parent, M. Damiani, J. de Macedo, F. Porto, and C. Vangenot. A Conceptual View on Trajectories. *Data Knowl. Eng.*, 65(1):126–146, 2008.
- [82] Cambridge Systematics. Data Collection in the Portland, Oregon Metropolitan Area: Case Study. Report DOT-T-97-09. U.S. Department of Transportation, 1996.
- [83] P. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Pearson Education, Inc., 2006.

- [84] R. Trasarti, M. Baglioni, and C. Renso. DAMSEL: A System for Progressive Querying and Reasoning on Movement Data. *International Workshop on Database and Expert Systems Applications*, 0:452–456, 2009.
- [85] R. Trasarti, S. Rinzivillo, F. Pinelli, M. Nanni, A. Monreale, C. Renso, D. Pedreschi, and F. Giannotti. Exploring Real Mobility Data with M-Atlas. In Balczar, Bonchi, Gionis, and Sebag, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6323 of *LNCS*, pages 624–627. Springer, 2010.
- [86] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna. Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art. *Journal of Web Semantics*, 4(1):14–28, 2006.
- [87] R. van Marwijk, H. Elands, and J. Lengkeek. Experiencing Nature: the Recognition of the Symbolic Environment within Research and Management of Visitor Flows. *Snow Landsc. Res.*, 81(1–2):59–76, 2007.
- [88] M. Vieira, P. Bakalov, and V. Tsotras. On-line Discovery of Flock Patterns in Spatio-Temporal Data. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS’09)*, pages 286–295, New York, NY, USA, 2009. ACM.
- [89] M. Wachowicz, R. Ong, C. Renso, and M. Nanni. Discovering Moving Flock Patterns among Pedestrians through Collective Coherence. Technical report, ISTI-CNR, Italy (to appear in the International Journal of Geographical Information Science), 2010.
- [90] J. Wolf. Using GPS Data Loggers to Replace Travel Diaries in the Collection of Travel Data. In *Dissertation, Georgia Institute of Technology, School of Civil and Environmental Engineering*, pages 58–65, 2000.
- [91] J. Wolf, R. Guensler, and W. Bachman. Elimination of the Travel Diary: An Experiment to Derive Trip Purpose from GPS Travel Data. *Transportation Research Record*, 1768:125–134, 2001.
- [92] Z. Wood and A. Galton. A Taxonomy of Collective Phenomena. *Applied Ontology*, 4(3-4):267–292, 2009.
- [93] World map, map of the world. <http://www.mapsofworld.com/>.
- [94] Z. Yan. Traj-ARIMA: a Spatial-Time Series Model for Network-constrained Trajectory. In *Proceedings of the 2nd International Workshop on Computational Transportation Science (IWCTS’10)*, pages 11–16, New York, NY, USA, 2010. ACM.
- [95] Z. Yan and S. Spaccapietra. Towards Semantic Trajectory Data Analysis: A Conceptual and Computational Approach. In *35th Very Large Data Base (VLDB) PhD Workshop*, 2009.
- [96] X. Zheng, T. Zhong, and M. Liu. Modeling Crowd Evacuation of a Building based on Seven Methodological Approaches. *Building and Environment*, 44(3):437–445, 2009.

- [97] Y. Zheng, L. Zhang, X. Xie, and W. Ma. Mining Interesting Locations and Travel Sequences from GPS Trajectories. In *Proceedings of the 18th International Conference on World Wide Web (WWW'09)*, pages 791–800, New York, NY, USA, 2009. ACM.