UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TOWARDS SEMANTIC ASSOCIATION RULES MINING FROM ONTOLOGY-BASED SEMANTIC TRAJECTORIES

ANTONIO CARLOS FALCÃO PETRI

ORIENTADOR: DIEGO FURTADO SILVA

São Carlos - SP

Fevereiro, 2021

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TOWARDS SEMANTIC ASSOCIATION RULES MINING FROM ONTOLOGY-BASED SEMANTIC TRAJECTORIES

ANTONIO CARLOS FALCÃO PETRI

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Inteligência Artificial.

Orientador: Diego Furtado Silva

São Carlos - SP

Fevereiro, 2021

Antonio Carlos Falcão Petri
Towards semantic association rules mining from ontology-based semantic trajectories/ Antonio Carlos Falcão Petri. – São Carlos - SP, Fevereiro, 2021-
131 p. : il. (algumas color.) ; 30 cm.

Orientador: Diego Furtado Silva

– Universidade Federal de São Carlos, Fevereiro, 2021.
1. Semantic Trajectory. 2. Data Mining. 3. Knowledge Base. I. Diego Furtado Silva. II. Universidade Federal de São Carlos. III. Departamento de Computação. IV. Towards semantic association rules mining from ontology-based semantic trajectories

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Defesa de Dissertação de Mestrado do candidato Antonio Carlos Falcão Petri, realizada em 09/02/2021.

Comissão Julgadora:

Prof. Dr. Diego Furtado Silva (UFSCar)

Profa. Dra. Marilde Terezinha Prado Santos (UFSCar)

Prof. Dr. Renato Fileto (UFSC)

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.
O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Ciência da Computação.

Este trabalho é dedicado a todas as pessoas para quem eu reclamei deste trabalho. Por estar cercado de boas pessoas, eu pude reclamar com muita gente e ainda assim ter todas elas me apoiando para chegar aqui.

Agradeço a Amandia, que me acompanhou nessa jornada acadêmica, desde o período em que eu tinha uma ideia por semana, até os períodos em que as ideias me faltavam. Também agradeço a ela por ter escolhido ser minha irmã.

Agradeço à minha família de São Carlos, composta pelo eloquente Muriel, o urso Zé, e o brother Aquino. Mesmo longe, eles estavam aqui.

Agradeço a Cissa, Henrique, Estudante e Sugaya por discutir todo e qualquer tipo de assunto comigo.

Agradeço a Giovanna por ser minha companheira durante todo esse tempo, o que por si só já resume várias responsabilidades e desafios a serem superados. Agradeço também a Andréa, por sempre acreditar e torcer por mim.

Agradeço aos meus pais e, nesse caso, não tenho motivos específicos a serem enumerados. Apenas os agradeço com toda intensidade e emoção que me preenche ao pensar neles.

Agradeço às centenas de episódios de podcasts que me fizeram companhia e me capacitaram para entender um pouco melhor a ciência, o mundo, a sociedade e as maravilhas e absurdos da vida.

Por fim, agradeço ao Diego, que começou a me orientar desde quando nos conhecemos pela primeira vez. Obrigado pelo bom exemplo de professor, pesquisador e orientador. Até logo, e obrigado pelos memes.

Acknowledgements

This work has been financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, by the Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil (CNPq) – Process 130790/2020-6, and by the grant #2017/24340-6, São Paulo Research Foundation (FAPESP).

Você estuda para errar menos, não para estar certo.
(Altay de Souza, Naruhodo)

Abstract

Different technologies and social-cultural aspects of our lives have allowed the acquisition of people's mobility data. The same applies to other moving objects, such as birds with GPS trackers and hurricanes with real-time satellite data. Although these raw positioning and timings are useful in many applications, it has been long recognized by the Trajectory Data community that semantics are required to capture the complexity of humans' and other objects' behaviors. Semantic Trajectories were proposed in this context as raw trajectories enriched with semantic annotations and possibly interlinked with external data. Based on these requirements, many works incorporate concepts and technologies from the Semantic Web to deal with the complexity of merging, representing, and querying heterogeneous data. They usually use ontologies to represent and manipulate concepts such as Moving Objects, Trajectories, Stops and Moves, and semantic aspects related to each of them. Nonetheless, we find that no previous work has explored mining patterns from these ontology-based representations. On the contrary, current efforts use standard association rule mining algorithms, such as Apriori, which require propositional data represented as Boolean feature vectors. To mine patterns aware of the semantic relations in a Semantic Trajectory ontology, we explore algorithms from the Knowledge Base Refinement field. These methods were proposed to use real-world facts represented in Knowledge Bases such as YAGO and DBPedia to infer new entities and relationships. We build on previous works describing ontology-based trajectory representations and tackle the knowledge discovery task using AMIE, a well-known state-of-the-art KB rule mining algorithm. This approach mines patterns in the form of Horn rules, which allows us to investigate associations between time, spatial, and semantic relations interlinking trajectory events. We show that representations previously proposed in the Semantic Trajectory community are not suitable to be directly mined by this approach. However, they can be easily extended to power the AMIE algorithm. We also describe and address different issues that arise when using a domain-agnostic mining algorithm. The proposed data pipeline mines interesting patterns in experiments using Foursquare datasets. Nonetheless, there is a large number of rules which state facts that are too general. We build on these issues and argue in favor of the design of a domain-specific mining algorithm. We discuss future opportunities based on the acquired experience and experiments. Our approach shows how the Semantic Trajectory and Knowledge Base Refinement communities have built in recent years a large number of representations and mining approaches that could be put together to mine rules with rich semantic expressiveness from semantic data.

Keywords: semantic trajectory. semantic data mining. association rule mining. knowledge base. ontology.

RESUMO

Diferentes tecnologias e aspectos socioculturais em nosso dia a dia permitem a aquisição de dados de mobilidade de pessoas. O mesmo se aplica a outros objetos móveis, como pássaros utilizando rastreadores GPS e furacões analisados via satélite. Embora estes dados de localização espacial e temporal sejam úteis em muitas aplicações, a comunidade de dados de trajetórias reconheceu há tempos a necessidade de aspectos semânticos para capturar a complexidade dos comportamentos de humanos e de outros objetos. Nesse contexto, foram propostas as Trajetórias Semânticas, que se baseiam em trajetórias espaço-temporais enriquecidas com anotações semânticas e possivelmente interligadas com dados externos. Por conta disso, muitos trabalhos incorporam conceitos e tecnologias da Web Semântica para lidar com a complexidade de representar, mesclar e consultar dados heterogêneos. Esses trabalhos geralmente utilizam ontologias para manipular conceitos como Objetos Móveis, Trajetórias, Paradas (*Stops*) e Movimentos (*Moves*), bem como os diferentes aspectos semânticos relacionados a cada um deles. Entretanto, não é possível encontrar na literatura trabalhos que explorem a mineração de padrões aplicada diretamente nestas representações baseadas em ontologia. Em vez disso, os esforços atuais utilizam algoritmos de mineração de regras de associação, como o *Apriori*, que requerem dados proposicionais. Esta dissertação explora algoritmos do campo do Refinamento de Bases de Conhecimento de modo a extrair padrões que tirem proveito das relações armazenadas em uma ontologia de Trajetórias Semânticas. A proposta original desses algoritmos é a inferência de novas entidades e relacionamentos em Bases de Conhecimento (*KBs*, do inglês *Knowledge Bases*), como a YAGO e a DBPedia, utilizando-se para isso os fatos já armazenados nas bases. Neste trabalho, utiliza-se a ferramenta AMIE, um representante do estado da arte na mineração de regras em *KBs*, que permite a extração eficiente de padrões na forma de Regras de Horn. No contexto de Trajetórias Semânticas, isso representa a mineração de associações entre as relações temporais, espaciais e semânticas que interligam eventos em uma base de trajetórias. O *pipeline* de dados proposto é capaz de extrair padrões interessantes em experimentos utilizando conjuntos de dados do Foursquare. No entanto, a utilização de um algoritmo agnóstico de domínio acaba por minerar um grande número de regras que definem fatos que são muito gerais. Construímos técnicas para avançar sobre essas questões e argumentamos a favor do desenvolvimento de um algoritmo de mineração específico de domínio. Além disso, a abordagem investigada mostra como as comunidades de Trajetórias Semânticas e Refinamento de *KBs* construíram um grande número de representações e abordagens de mineração que poderiam ser reunidas para extrair padrões com rica expressividade semântica a partir de dados semânticos.

Palavras-chave: trajetória semântica. mineração de dados semânticos. mineração de regras de associação. base de conhecimento. ontologia.

LIST OF ABBREVIATIONS AND ACRONYMS

ARM	Association Rules Mining
CWA	Closed World Assumption
DL	Description Logics
FOI	Feature of Interest
FIM	Frequent Itemset Mining
ILP	Inductive Logic Programming
KB	Knowledge Base
KDD	Knowledge Discovery in Databases
KG	Knowledge Graph
LBSN	Location-Based Social Networks
LOD	Linked Open Data
MRDM	Multi-relational Data Mining
OWA	Open World Assumption
OWL	Web Ontology Language
POI	Point of Interest
RDF	Resource Description Framework
RDM	Relational Data Mining
SPARQL	SPARQL Protocol and RDF Query Language
SW	Semantic Web
W3C	World Wide Web Consortium

List of Figures

List of Tables

List of Algorithms

CONTENTS

Chapter 1
Introduction

Consider a typical workday in a big city. As the day begins, many people start their daily routines, moving from one place to another using different transportation modes. Night-workers might go home in the morning, while most people will be going from home to work. Maybe some will drop off their kids at school and take them on their way back home at the end of their working day. Supermarkets and restaurants may be visited during this period. Tourists visiting the city can be expected to go from a hotel to a tourist place, and a person on a business trip may go from a hotel to a workplace.

Interestingly, different technologies and social-cultural aspects of our lives have allowed the acquisition of people mobility's data. For example, carrying mobile phones in our pockets is usually equivalent to having a GPS tracker generating a stream of positions throughout our day. The same applies to other moving objects, such as birds with GPS trackers and hurricanes with real-time satellite data. The sequence of time-stamped geolocations yields a *raw trajectory*, such as the one depicted in Figure 1.1.

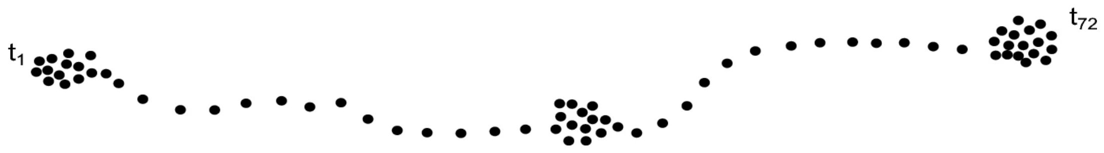


Figure 1.1 – Example of a raw trajectory obtained by sampling a user's geo-location at different timestamps. Each black dot represents a sample, and their clusters show locations where the user stayed for a while. Source: Ferrero et al. (2020).

Many challenges arise when dealing with tracking technologies and trajectory data. Nonetheless, the knowledge hidden in this data and its potential to impact real-world problems have promoted the emerging of many research fields, such as *Behavior informatics* (RENSO et al., 2013), Computational *Movement Analysis* (LAUBE, 2014), and *LBSN-enriched Urban Computing*[1] (SILVA et al., 2019).

Much of the interest in trajectory data is related to finding patterns such as mobility behaviors and objects' preferences. The pattern discovery task is powered by Data Mining, which

[1] LBSN: Location-based Social Network, as will be defined next.

is the research field interested in uncovering hidden and potentially useful information from data. Data mining can also be described as the process to transform data into knowledge (FAYYAD et al., 1996), being sometimes called knowledge discovery or Knowledge Discovery in Databases (KDD).

When first massive volumes of spatiotemporal trajectory data became available, data mining was applied to answer questions such as *"Can we cluster trajectories into a small set of representative groups?"*, *"Can we find regions with a high number of visits?"*, and *"Are there frequent sequences of locations visited by the moving objects with similar travel times?"* (ATLURI et al., 2018).

While such patterns exist, many factors contribute to small or significant changes in objects' mobility behavior. For example, consider how people change their routine depending on the weather or road work in their itinerary.

Nowadays, it has been long recognized by the trajectory data community that only raw positions and timings are not enough to capture the complexity of humans' and other objects' behaviors. *Semantic Trajectories* were proposed as raw trajectories enriched with semantic annotations, and possibly interlinked with external data (ALBANNA et al., 2015).

The term *Semantic Trajectory Data Mining* refers to the data mining process applied to Semantic Trajectories. For example, a trajectory or a trajectory segment may be annotated with the user's goal or behavior (BOGORNY et al., 2014). Geo-located photos published by a tourist in a social network may be used to understand what the user sees and predict their interests and next places (TAKIMOTO et al., 2017). People's interactions in a public park may be inferred by analyzing their trajectories (RENSO et al., 2013).

Figure 1.2 shows an example of a semantically enriched trajectory. Besides the raw trajectory data previously shown in Figure 1.1, many other semantic elements capture different aspects of someone's trajectory. For example, we have information about how they moved from one place to another, about the visited places' category, and even messages in multiple social media platforms that may be used as a proxy to how the person was feeling at each moment (FERRERO et al., 2020).

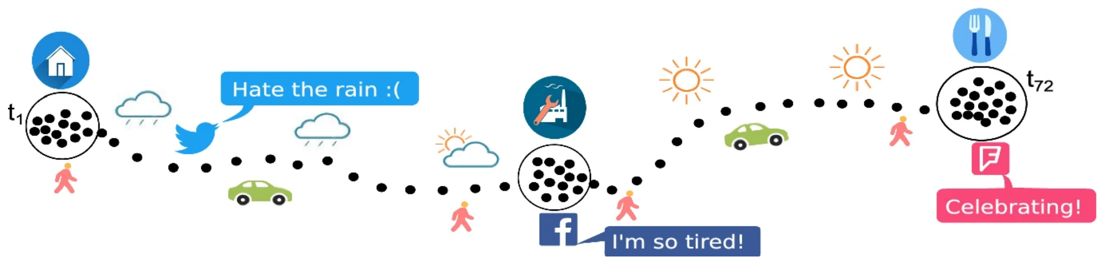


Figure 1.2 – Example of a semantic trajectory obtained by processing data from Figure 1.1 and integrating it with external sources. Data such as weather, social media messages, transportation means and place identification help to capture the context in which the trajectory occurred. Source: Ferrero et al. (2020).

Semantic Trajectory requires the integration of raw trajectory with any other useful external data. Many proposed approaches incorporate concepts and technologies from the Semantic Web to deal with the complexity of merging, representing, and querying heterogeneous data.

The *Semantic Web* (SW) was proposed almost two decades ago as the next step of the World Wide Web. In the SW, web pages would not just present data to users but use a set of tools to serve semantically annotated information in a machine-interpretable format. Different systems would communicate by accessing knowledge distributed on the whole web and formal logic would be used to infer new knowledge (Berners-Lee et al., 2001). Ontologies are the formal knowledge representation used.

The Semantic Web can be associated with Semantic Trajectories in two ways. Firstly, SW can provide concepts and tools to represent and process knowledge. Semantic Trajectories can, therefore, be represented with SW techniques, where data can be stored using a formal schema and annotated with semantics, constraints, and rules, allowing knowledge checking and inference.

Secondly, Semantic Trajectories can be further enriched with the knowledge contained in LOD databases. The term *Linked Open Data* (LOD) is used to refer to a growing collection of publicly available interlinked datasets represented in the SW's machine-interpretable format (BIZER et al., 2009). For example, the public LOD database LinkedGeoData[2] can be used to annotate a user's visited-place with public knowledge, such as an address, phone number, and category (FILETO et al., 2015b).

Many technologies have been developed to support the Semantic Web and then used to create LOD data. Semantic Web and Linked Open Data have also inspired the building of huge Knowledge Bases (KBs), which store structured facts about real-world entities and concepts. DBpedia (AUER et al., 2007), YAGO (SUCHANEK et al., 2007), and Wikidata (VRANDEČIĆ; KRÖTZSCH, 2014) are examples of knowledge bases build semi-automatically from Wikipedia[3] and other web resources.

Many tasks and algorithms were investigated in the context of building and extending knowledge bases. Initial work focused on acquiring and extracting knowledge from web pages and representing it in a well-structured format. Once knowledge bases acquired enough knowledge, *knowledge bases refinement* methods were proposed to use the learned knowledge to infer new entities and relations (PAULHEIM, 2016). One approach is to mine frequent logical rules that can be subsequently used to infer and extend KB's facts (GALÁRRAGA et al., 2013).

In this dissertation, we are interested in mining Association Rules from Semantic Trajectory data. More specifically, we are interested in mining Horn rules (LAJUS et al., 2020), which are more expressive than conventional, transaction-based association rules (AGRAWAL et al.,

[2] <http://linkedgeodata.org>
[3] <https://wikipedia.org>

1993). We borrow from the Knowledge Base refinement task a state-of-the-art algorithm and explore how it can be applied to mine *logical rules* from an ontology-based semantic trajectory representation.

1.1 Motivation and examples

Mined rules can be used in different ways. In the context of trajectory data, they can communicate hidden mobility patterns, behaviors, and preferences, as well as to predict data, such as a user's next place. In general, they can be used as an automatic step to retrieve potentially interesting patterns in a knowledge discovery framework.

One example of a logical rule that can be mined with the approach proposed in this work is given by Rule 1.1[4].

$$\begin{array}{ll} \text{if} & visit\ V1\ \textbf{at}\ The\ Orion\ Penthouse \\ \text{and} & visit\ V2\ \textbf{during}\ Afternoon \\ \text{and} & \quad V2\ \textbf{within 2 kilometers of}\ V1 \\ \text{then} & \quad V2\ \textbf{is at}\ TimesSquare \\ \text{with} & \text{confidence}\ 53\% \end{array} \tag{Rule 1.1}$$

This rule states that given a visit to the *The Orion Penthouse*[5] (a residential building in New York City) and a nearby place (within 2 kilometers) during the afternoon, this nearby place is usually (53 percent of the time) the Times Square[6]. Rule 1.1 uses different semantics (the venue's name/identifier and the visit's time of the day) and a relation between the two visits (within 2 kilometers) to describe an user's mobility behavior.

This relationship-awareness contrasts with current works in literature, which cannot explore arbitrarily-defined relations between entities. Indeed, works such as Bogorny et al. (2009), Rizk and Elragal (2012), Mousavi et al. (2016), Khoshahval et al. (2017) use standard association rule mining algorithms, such as Apriori (AGRAWAL et al., 1994), which require propositional data represented as feature vectors. Converting from relational into propositional representations is done in a process called *propositionalization*, but comes with the cost of limited expressiveness regarding concepts' relations (RISTOSKI; PAULHEIM, 2016; DŽEROSKI, 2009).

Being able to mine relational data is especially interesting since many recent works use ontology-based representations to model semantically-enriched trajectory data (RENSO et al., 2013; FILETO et al., 2015b; NOGUEIRA et al., 2018; MELLO et al., 2019). The graph-like representation used by ontologies allows representing interesting relations between concepts such as the friendship between users (MELLO et al., 2019).

[4] As previously mined in Petri and Silva (2020).
[5] <https://foursquare.com/v/the-orion-penthouse/4f93f1c8e5e828f50a2b81d1>
[6] <https://foursquare.com/v/times-square/49b7ed6df964a52030531fe3>

Different data mining techniques can be borrowed and adapted from multiple research areas and then used to leverage rule mining in ontology-based representations. More specifically, we focus on learning rules using algorithms tailored to ontology-based data representations.

Next, we show some examples of rules similar to those that we aim to mine in this work. Here, we present them in natural language but following some structure similar to the formalism used to mine the rules. Whether we can mine them depends on multiple factors, such as the data representation strategy and the mining algorithm used. These factors will be appropriately discussed in this dissertation.

Consider Rule 1.2. It states that if two people, lets say **A**lice and **B**ob, visit a venue which is somebody's home, then **A**lice and **B**ob are possibly friends.

$$\begin{array}{ll} \text{if} & user\ a\ \textbf{visited}\ venue\ v \\ \text{and} & user\ b\ \textbf{visited}\ venue\ v \\ \text{and} & v\ \textbf{has category}\ Home \\ \text{then} & user\ a\ \textbf{is friend of}\ user\ b \\ \text{with} & \text{confidence } 90\% \end{array} \tag{Rule 1.2}$$

Rule 1.3 states that visiting a place shortly after or before visiting the venue **A**, usually means visiting the venue **B**. For example,

$$\begin{array}{ll} \text{if} & visit\ v1\ \textbf{at}\ venue\ A \\ \text{and} & visit\ v2\ \textbf{within 2 hours of}\ v1 \\ \text{then} & v2\ \textbf{at}\ venue\ B \\ \text{with} & \text{confidence } 53\% \end{array} \tag{Rule 1.3}$$

Rule 1.4 is somewhat similar to Rule 1.3, and shows that if someone was at an Office and is now in a nearby place, he or she can be inferred to be probably in a Restaurant.

$$\begin{array}{ll} \text{if} & visit\ v1\ \textbf{at}\ Office \\ \text{and} & visit\ v2\ \textbf{at}\ venue\ x \\ \text{and} & v1\ \textbf{before}\ visit\ v2 \\ \text{and} & v2\ \textbf{within 2 kilometers of}\ v1 \\ \text{then} & x\ \textbf{is}\ Restaurant \\ \text{with} & \text{confidence } 85\% \end{array} \tag{Rule 1.4}$$

Although outside of this research's scope, it would also be interesting to learn patterns such as Rule 1.5, where some venues' distance threshold (k) is automatically chosen to maximize the rule's confidence. Indeed, this could be accomplished by implementing and extending ideas from Galárraga and Suchanek (2014) to mine numerical rules.

$$\begin{array}{ll} \text{if} & visit\ v1\ \textbf{at}\ Office \\ \text{and} & visit\ v2\ \textbf{at}\ venue\ x \\ \text{and} & v1\ \textbf{before}\ visit\ v2 \\ \text{and} & v2\ \textbf{within}\ k\ \textbf{kilometers of}\ v1 \\ \text{and} & k < 2 \\ \text{then} & x\ \textbf{is}\ Restaurant \\ \text{with} & \text{confidence } 95\% \end{array} \qquad \text{(Rule 1.5)}$$

In a general sense, the idea of representing data as an ontology and mining association rules can be extended to other domains. What is required is that we consider what kind of relations would be easily represented in an ontology but hardly represented in a propositionalized form. Also, there must be an interest in explicitly representing such relations.

1.2 Proposal

In this work, we propose to use KB rule learning algorithms as general-purpose pattern mining tools. Moreover, we identify Semantic Trajectories as a potential domain to apply this approach. We build on previous works describing ontology-based trajectory representations and tackle the knowledge discovery task by mining logical association rules, expressed as Horn rules.

This approach allows us to investigate associations between time, spatial, and semantic relations interlinking trajectory events. The explored rule mining strategy can use these relations to capture interesting mobility patterns and users' preferences.

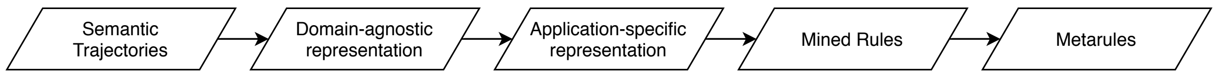
Therefore, this work is based on two main hypotheses: (i) KB rule learning algorithms can mine interesting patterns when applied to ontology-based data, even when such data is not traditionally considered to be a KB; (ii) Semantic Trajectory is a suitable domain to test the first hypothesis.

To the best of our knowledge, mining association rules directly from the ontology-based trajectory representation has not been investigated by the trajectory community yet. Therefore, our first approach to mine rules is based on AMIE, a well-known state-of-the-art KB rule mining algorithm.

To mine and interpret logical rules from ontology-based semantic trajectories, we consider a 5-step data pipeline depicted in Figure 1.3. The five steps are as follows:

- **Semantic Trajectories:** This first data step includes all data pre-processing, cleaning, anonymization, and semantic enrichment required to deliver semantic trajectories data.

Figure 1.3 – The proposed data pipeline for logical rule mining.


Semantic Trajectories
Domain-agnostic representation
Application-specific representation
Mined Rules
Metarules

Multiple works discuss how to generate, acquire, or build semantic trajectories. In our experiments, we use publicly available semantic trajectories datasets from Location-based Social Networks (LBSN) (CHO et al., 2011; Dingqi Yang et al., 2015; YANG et al., 2019; YANG et al., 2020).

- **Domain-agnostic ontology representation:** Multiple ontology-based representations have been proposed in the literature (FILETO et al., 2015b; NOGUEIRA et al., 2018; MELLO et al., 2019). We have chosen a suitable representation in this data step and populated it with all data from the previous one (NOGUEIRA et al., 2018).
- **Application-specific ontology representation:** The first main contribution of this research is proposing a systematical approach for deriving a dynamic application-specific representation. We show that this representation is more suitable for being mined by a relational mining algorithm.
- **Mined Rules:** We apply an off-the-shelf general-purpose mining algorithm to mine patterns from our data ontology-based representation (LAJUS et al., 2020). We consider multiple aspects, such as pruning and interestingness metrics, and then extend the algorithm to better deal with our application-domain.
- **Meta-rules:** We propose an approach to group similar mined rules based on the concept of meta-rules (KAMBER et al., 1997; DJENOURI et al., 2013).

We apply AMIE to location-based social network datasets and use them to discuss issues on using an off-the-shelf algorithm, as well as the opportunities in designing a domain-tailored rule mining algorithm. As discussed in the following chapters, this represents a step towards more exciting and contextual association rules, allowing new insights on trajectory data to be drawn.

1.3 Objectives

The main objective of this dissertation is to validate the hypothesis that KB rule learning algorithms may be applied to data not traditionally viewed as a KB by using Semantic Trajectories as a case study.

As secondary objectives, we can enumerate:

- Mine association rules from semantic trajectories that capture interesting users' behaviors;

- Direct future research on applying KB mining algorithms in ontology-based data, especially Semantic Trajectory domain;
- Adapt the AMIE algorithm towards working with a broader set of mining strategy requirements;
- Investigate the challenges that arise when converting trajectories data to an application-specific representation;
- Contribute to open-source projects used throughout the dissertation;
- Publicly share the technical artifacts developed in this research, including code and experiments within a reproducible environment.

1.4 Scope

As previously discussed, we would like to validate with this work that KB rule mining algorithms can be applied to general domains. Nonetheless, we need to limit the research's scope in order to make it feasible.

We focus on Semantic Trajectory Data Mining as an appealing domain to apply these methods. The reasons for this choice are twofold: firstly, research in Semantic Trajectory has already developed datasets, representations, and algorithms that can serve as the basis for the present work; secondly, the rules mined by this approach have great potential to capture interesting trajectory patterns.

Given the multiple ontologies proposed in the literature for representing trajectory data (see Section 2.3), we focus on the STEP ontology, which is publicly available and well-documented.

Besides the application domain and data representation, we also limit the scope of the KB rule learning algorithms explored and the Semantic Trajectory dataset investigated. We focus on applying the AMIE algorithm for learning association rules and complement the discussion by citing other algorithms and approaches (see Section 2.4).

To validate the hypotheses, we choose different datasets from location-based social networks that inherently contain semantic data and can be further integrated with external knowledge.

1.5 Limitations

As an initial step towards mining logical rules from trajectory data, this work has an important set of limitations. Although also discussed in Chapter 5, we summarize them here:

- This dissertation proposes to apply an off-the-shelf domain-agnostic mining system, AMIE, to a highly specialized domain, Semantic Trajectories. Consequently, it imposes a clear limitation: the algorithm does not consider the domain knowledge that could guide the mining process towards more interesting patterns. This work starts the discussion on how to design better systems for specific domains.

- We find that an application-specific ontology representation is required to mine interesting rules. Although we propose an automatic process to build it, we note that this tightly couples the initial data representation to the data mining algorithm.

- AMIE has low support for ontology schema. Therefore, we cannot mine patterns using, for example, a venue's category taxonomy (BOGORNY et al., 2009).

- A domain-expert does not validate the proposed mining process and mining results. As will be further discussed, the previously described topics impose a strict limitation on the mined rules' direct usability. Nonetheless, we show that the proposed mining pipeline and mining strategy have exciting characteristics and could be refined in future works.

1.6 Main contributions

In this section, we summarize the contributions made by this dissertation. We highlight that they represent a step towards mining context-aware relation-centric rules from Semantic Trajectories and possibly other domains. Part of these contributions have also been published in the paper Petri and Silva (2020) at the *IEEE ICMLA 2020 Conference*[7].

- Investigate the usage of a KB rule learning algorithm on mining rules from an ontology-based representation of Semantic Trajectories;

- Investigate the expressiveness power of the mined rules when capturing semantic trajectories patterns;

- Contribute to directing future research on applying KB mining algorithms in ontology-based data;

- Mine association rules from semantic trajectories that capture mobilities' behaviors;

- Contribute to the Trajminer code library (PETRY et al., 2019b), which contains the implementation of many trajectory manipulations and algorithms;

- Contribute to AMIE 3 code by collaborating with the original authors, investigating performance bottlenecks, and adding new features to the framework;

[7] <https://www.icmla-conference.org/icmla20/>

- Publicly share the technical artifacts used in this research, such as code and software requirements, to allow the reproducibility and extension of this work.

1.7 Dissertation outline

The remainder of this dissertation is organized as follows:

We discuss the foundation topics of this work in Chapter 2. They include the basic concepts related to: (i) *Ontologies* and *Knowledge Bases*, (ii) *Association Rule Mining*, (iii) *Semantic Trajectory*, from their modeling to pattern mining, and (iv) approaches for *mining rules in Knowledge Bases*.

We proceed to discuss the proposed vision of using KB mining techniques to ontology-based semantic trajectories in Chapter 3. Chapter 4 discusses multiple experiments and data analysis we explored in this work.

Finally, Chapter 5 summarizes this dissertation, including its contributions and hypothesis results. The limitations of the current work are explored, and an extensive range of research and development opportunities are discussed.

Complementary, we briefly describe in Section 6.1 the technical stack used to implement this work. Extra experimental data from Chapter 4 is provided in Section 6.2.

Chapter 2
THEORETICAL FOUNDATIONS

This dissertation's scope spans a multitude of research fields and sub-fields. In this chapter, we review the main building blocks of this work:

- Ontology, Semantic Web and Knowledge Bases (Section 2.1): how heterogeneous and exogenous data can be represented, and how knowledge has been explicitly and formally structured to support different applications;
- Association Rule Mining (Section 2.2): what are the tasks of Frequent Itemset Mining and Association Rule Mining.
- Semantic Trajectories (Section 2.3): what are Semantic Trajectories, how they have been modeled as Ontologies, and how data mining has been applied to power multiple applications.
- Learning rules from Knowledge Bases (Section 2.4): how to efficiently extract patterns from huge volumes of data represented as Knowledge Bases.

2.1 Ontology, Semantic Web, and Knowledge Bases

In many domains, it is useful or even necessary to represent knowledge formally. For example, many successful knowledge formalizations exist in biology and medicine, such as the ones in the Gene Ontology (GO) project. GO formalizations use a specific set of terms to describe different aspects of genes and genes product functionality (du Plessis et al., 2011).

Knowledge Representation and Reasoning is a sub-field of Artificial Intelligence interested in representing information in a machine-understandable and -processable form. *Ontologies* are a successful representation format, usually defined as a formal and explicit specification of a conceptualization (GRUBER, 1993).

An ontology can provide a general view of the world, being referred to as a *top-level ontology*, or model a specific domain, being called a *domain ontology*. Other terms may be used, such as *application ontology* for an ontology that is defined to serve a specific task in a given domain.

Ontologies are described by languages. The Web Ontology Language[1] (OWL) is the most well-known standard and a W3C[2] recommendation. OWL is based on Description Logics (DL), which is a subset of first-order logic (CHUANGLU, 2012).

DL and ontologies allow the representation of classes (concepts), roles (properties), individuals (instances), and axioms. The latter are expressions that can be used to represent inference rules.

An OWL ontology is a set of facts. These facts can be partitioned into two subsets, called T-Box and A-Box. The first one contains the facts that define classes and classes hierarchy, and domains and ranges for predicates. The second one contains instance data (GALÁRRAGA et al., 2013).

OWL also comes with many reasoners. The main tasks provided by these ontology inference engines are: (i) to check whether the ontology is consistent (consistency checking), (ii) to find whether a concept is subsumed by another one (subsumption), and (iii) to check what classes a given instance belongs to (instance checking) (ABBURU, 2012; RENSO et al., 2013).

The Resource Description Framework (RDF) (W3C, 2014) is a format for information exchange used by OWL to implement its Description Logics capabilities. RDF is a data model that allows the specification of triples in the form of *subject-predicate-object*. A triple represents that there is a relation (given by the *predicate*) between the *subject* and the *object*.

A set of RDF triples can be viewed as a graph, where *subject* and *object* are nodes, and the *predicate* is a directed link connecting them. This may be referred to as an RDF-graph (LAMY, 2017).

Ontologies can be queried using query languages. SPARQL (SPARQL Protocol and RDF Query Language) is the most used one, being usually modified to specific cases, generating other query languages (BARBIERI et al., 2010; PERRY; HERRING, 2012).

Although ontologies can also represent instances, they are sometimes (erroneously) used to describe only the data schema aspect (EHRLINGER; WÖSS, 2016). Therefore, it is common to use the term *ontology* when the *data schema* is more important than the data itself. On the other hand, the term *Knowledge Base* is usually used when the *data* is more important than its schema. Besides the context in which the terms are used, Ehrlinger and Wöß (2016) consider that there is no difference between Ontologies and Knowledge Bases.

Both ontologies and KBs are old terms that became popular again after the *Semantic Web* (SW) was proposed in 2001 as the future of the World Wide Web (Berners-Lee et al., 2001; SUCHANEK et al., 2019). In the proposed view, knowledge would be shared on the web, used

[1] <https://www.w3.org/TR/2012/REC-owl2-overview-20121211/>
[2] World Wide Web Consortium

by intelligent agents, and integrated with different semantic data sources. The Semantic Web did not happen on the predicted scale but stimulated much research and the development of a stack of specifications and tools. OWL, RDF, and SPARQL are technologies that were proposed in the context of the SW.

Years later, in 2009, the term *Linked Open Data* was used to refer to a variation of Semantic Web. Linked Open Data refers to a publicly available interlinked collection of datasets (BIZER et al., 2009).

Ontologies are a fundamental component to Semantic Web and Linked Open Data. They allow interoperability between different systems, as well as the specification of inference rules.

Complementary, the literature has commonly used the term KB to talk about private and public projects focused on building structured data to serve multiple tasks, such as intelligent web search, question understanding, social media mining, and biomedicine (RISTOSKI; PAULHEIM, 2016; OMRAN et al., 2018; SUCHANEK et al., 2019).

Examples of publicly available KBs include Freebase (BOLLACKER et al., 2008) and Wikidata (VRANDEČIĆ; KRÖTZSCH, 2014), which are KBs edited by the crowd, and YAGO (SUCHANEK et al., 2007) and DBpedia (AUER et al., 2007) which extract knowledge by consuming semi-structured data from pages like Wikipedia. NELL ontology, from the project Never-ending Language Learner (MITCHELL et al., 2018), represents another approach for KB construction, as it combines many data mining and machine learning models to extract knowledge from unstructured web pages.

KBs may model general concepts such as people, countries, cities, movies, and animals. Furthermore, each instance of these concepts might contain associated data, such as a person's name or a movie's director. Instances of concepts might also be interrelated by relations such as `director of`, connecting a `Person` and a `Movie`. Current KBs contain millions of entities and hundreds of millions of facts (GALÁRRAGA et al., 2013; PAULHEIM, 2016).

More recently, the term *Knowledge Graph* (KG) has also been used both commercially and in academia to refer to Knowledge Bases that have an underlying representation of a graph. Knowledge Graph was a term initially used by Google back in 2012 to refer to an internal knowledge base that was going to be used to enrich services like Google Search[3] (PAULHEIM, 2016). Companies such as Yahoo!, Microsoft, and Bloomberg also have projects on commercial knowledge bases (MITCHELL et al., 2018).

These terms are many times used without a proper definition or consensus (EHRLINGER; WÖSS, 2016). Following Paulheim (2016), we do not try to define a Knowledge Base or Knowledge Graph formally, but instead, list some general characteristics:

- they mainly describe real-world entities and their interrelations;

[3] <http://googleblog.blogspot.co.uk/2012/05/introducing-knowledge-graph-things-not.html.>

- they define possible classes and relations of entities in a schema;
- they allow for potentially interrelating arbitrary entities with each other;
- they cover various topical domains.

For a more complete and philosophical definition of Knowledge Bases, we refer the reader to Suchanek et al. (2019).

As the data sources are noisy and far from being complete, many approaches have been proposed to clean and extend the knowledge represented. These approaches have been more commonly grouped in the *knowledge graph refinement* task. This topic will be discussed in Section 2.4.

2.2 Association Rule Mining

In this dissertation, we are mainly interested in the Association Rule Mining (ARM) task. Moreover, we are interested in mining knowledge from an ontology-based representation of semantically enriched data.

This section reviews the original association rule mining problem and the base algorithm *Apriori*. We delegate to Section 2.3 the discussion about ARM in the context of spatiotemporal and trajectory data and to Section 2.4 the discussion about ARM in the context of more complex data, i.e., Knowledge Bases, and more complex rules, such as multi-relational rules.

Association rule mining is a data mining task originally proposed in Agrawal et al. (1993) to find regularities in the co-occurrence of items in a database (RAMEZANI et al., 2014). The initial applications were on market basket analysis, where the goal was to detect items commonly bought together. Association rule mining usually operates on a set of records (transactions), where each record contains a set of items (SRIKANT; AGRAWAL, 1996).

The co-occurrence of items can be used to infer association, correlation, or some informal structure. Association rules capture knowledge by finding a relationship in the form of a set of items A implying in a set of items B, denoted as $A \rightarrow B$ (SRIKANT; AGRAWAL, 1996). This rule captures the information that transactions containing the items in A also tend to contain the items in B (SRIKANT; AGRAWAL, 1997). In this case, A is called the rule's antecedent, and B is called the consequent. A numeric value can be associated with the rule indicating, for example, the frequency in which the rule is true in the given dataset.

In this section, we use the formal definitions given in Fournier-Viger et al. (2017). Let $I = \{i_1, ..., i_m\}$ be a set of items (symbols). A transaction database $D = \{T_1, ..., T_n\}$ is a set of transacations, where $T_q \subseteq I (1 \leq q \leq n)$, and each T_q has a unique identifier q, called its Transaction Identifier. An itemset X is a set of items such that $X \subseteq I$. An association rule is an implication in the format $A \rightarrow B$, where A and B are itemsets and $A \cap B = \emptyset$.

ARM is a two-step process: firstly, a list of all the frequent sets of items is generated in a task known as *Frequent Itemset Mining* (FIM); secondly, each set is used to spawn many association rules between its items. The latter step is trivial and is not very sophisticated (AGGARWAL et al., 2014). Nonetheless, post-processing these rules is an interesting task. For example, the rules can be pruned, queried, or filtered using domain knowledge, possibly encoded as an ontology (DOU et al., 2015).

The difference between different ARM approaches and algorithms are usually in the FIM task, which is responsible to discover all frequent itemsets in a database. Generating these frequent itemsets can be further divided into two sub-problems: generating the potential set of frequent items, called *candidate itemsets*, and filtering them by different metrics, generating what is called as *frequent itemsets*. The different algorithms usually are variations on how the search space is explored (AGGARWAL et al., 2014).

Different metrics can be used to classify an itemset as frequent and to represent an interestingness/goodness measure of each association rule. Classically, *support* is the metric used to quantify how frequent an itemset is. Support (or absolute support) of an itemset X is defined as $sup(X) = |\{T|X \subseteq T \wedge T \in D\}|$, i.e., the number of transactions in a database D that contain all items in X. A user-specified parameter *minsup* is used as a threshold. An itemset with support greater than this threshold is said to be frequent.

Accordingly, the classical definition of goodness of a rule, called confidence, can be defined as $conf(A \rightarrow B) = \frac{sup(A \cup B)}{sup(A)}$. A rule's confidence can be interpreted as the probability of consequent B occurring conditioned on the antecedent A occurrence. Different domains, data, and applications require different metrics. We refer the reader to Geng and Hamilton (2006) for a complete review of metrics in data mining and association rule mining.

Algorithm 2.1 shows a baseline/prototype FIM algorithm defined in Aggarwal et al. (2014). It takes as input a transaction database T, and a user-defined support threshold $minsup$. Firstly, it generates all frequent patterns of length one, i.e., all frequent patterns that contain one item. An iterative process is started, using the previously mined frequent patterns to generate new candidate patterns. These candidate patterns are filtered using the support metric, and the selected patterns are added to the set of mined frequent patterns. The process continues until no new frequent pattern is found.

There are two key issues related to the computational efficiency of an FIM algorithm. Firstly, the generation of candidate patterns must be done in an orderly and carefully designed fashion, pruning irrelevant and duplicated candidates. Secondly, checking if a candidate pattern is a frequent pattern must use strategies to avoid looking to the full database at every step (AGGARWAL et al., 2014).

One well-known classical ARM algorithm is Apriori, which was the first algorithm proposed to mine association rules (AGRAWAL et al., 1994; AGGARWAL et al., 2014). Apri-

Algorithm 2.1 Baseline FIM. Adapted from Aggarwal et al. (2014).

```
1: procedure BASELINE FREQUENT ITEMSET MINING(DATABASE: $T$, MINIMUM SUPPORT: $minsup$)
2:     $FP \leftarrow$ length-one frequent patterns
3:     repeat
4:         Generate a candidate pattern $P$ from one (or more) frequent pattern(s) in $FP$
5:         if $support(P, T) >= minsup$ then
6:             Add P to frequent pattern set $FP$
7:     until all frequent patterns in $FP$ are explored
```

ori explores the space of candidate patterns using a breadth-first approach and considering a structured arrangement of itemsets. Candidate patterns with the length of $k + 1$ are generated by joining pairs of frequent patterns of length k.

The algorithm uses the *downward closure property* to prune rules before calculating their support. The *downward closure*, known as the *Apriori pruning trick*, is a frequent pattern property according to which every subset of a frequent pattern is also frequent (AGGARWAL et al., 2014). In Apriori, this means that every subset of a candidate pattern must be an already mined frequent pattern.

Once all frequent patterns have been mined in FIM, Apriori generates all possible association rules from each itemset. The rules are filtered using their confidence values and can be further pruned in a post-processing step.

The definition of the FIM task can be generalized to a broad range of domains and data. Consequently, the ARM task can also be applied. As discussed in Fournier-Viger et al. (2017), a customer transaction database can be generalized as a database of instances describing objects (the transactions), and each object is described using nominal attribute values (the items). Therefore, FIM can be defined as the task of finding attribute values that frequently co-occur in a database.

Ontology-based association rule mining is usually related to using ontologies to provide constraints. In this case, domain knowledge is modeled as an ontology and can be used to constrain the search space during the pattern mining task or post-processing to filter out inconsistent rules. The Semantic Web can also be used to enrich the data in the early steps (DOU et al., 2015; RISTOSKI; PAULHEIM, 2016).

We refer the reader to Fournier-Viger et al. (2017) and Aggarwal et al. (2014) for surveys on the FIM task and its variations, and to Dou et al. (2015) and Ristoski and Paulheim (2016) surveys on how ontologies and the Semantic Web can be used in ARM.

Many performance improvements to FIM and ARM tasks are continuously proposed in the literature. Also, there are many variations to the original FIM problem. For example, some algorithms are designed to operate on non-static databases, i.e., databases where transactions can be inserted, removed, or modified (Fournier-Viger et al., 2017).

Another interesting FIM variation is to mine multi-level association rules, as initially proposed in Srikant and Agrawal (1997). In this case, items can be arranged in a user-defined taxonomy (is-a hierarchies), and rules with different semantic granularities can be mined, such as $milk \rightarrow bread$, and $diary\ product \rightarrow bakery\ products$ (DOU et al., 2015).

Another variation of FIM is sequential pattern mining. In this case, the task is to discover sequences that frequently appear in a database that stores a sequence of transactions (Fournier-Viger; LIN, 2017).

After surveying the state-of-the-art on FIM, Fournier-Viger et al. (2017) lists some research opportunities. In this work, we tackle the following ones:

- Novel applications: there is an opportunity to apply existing pattern mining algorithms in new application domains, such as Social Networks and the Internet of Things. We do so in the domain of Semantic Trajectories;
- Complex data: another opportunity is to extend pattern mining algorithms to consider more complex data. Fournier-Viger et al. (2017) use as an example the task of mining spatial patterns, which will be discussed in Section 2.3. We do mine more complex data by using an ontology-based representation of semantic trajectories;
- Complex patterns: extending pattern mining algorithms to discover more complex and meaningful types of patterns. We do so by using algorithms capable of exploring the rich semantics represented in the proposed ontology-based semantic trajectory data.

2.3 Semantic Trajectory

Many technologies generate or have been applied to generate data about the mobility of people, vehicles, animals, and natural phenomena (ZHENG, 2015). They include Global Positioning Systems (GPS), Radio Frequency Identification (RFID), smartphone sensors, social networks and so on (PARENT et al., 2013; SILVA et al., 2019).

The dissemination of these technologies in the last decades allowed the easy and low-cost acquisition of *Trajectory Data*, a type of *Spatio-Temporal Data* that represents sequences of timestamped locations of moving objects (FENG; ZHU, 2016; ATLURI et al., 2018). Motivated by this data availability, many works started to investigate the challenges and the applications of huge volumes of trajectory data.

As defined in Feng and Zhu (2016), a trajectory is generated by sampling a moving object trace as it moves in space. Formally, a trajectory $\mathcal{T}$ is a time-sorted sequence of timestamped positions, i.e., $\mathcal{T} = \langle p_1, t_1 \rangle, \langle p_2, t_2 \rangle, ..., \langle p_n, t_n \rangle$, where $\langle p_i, t_i \rangle$ represents that the moving object was at position p_i at moment t_i. Also, the positions are ordered in time, i.e., $t_i < t_j, 1 \leq i < j \leq n$. A trajectory position is usually a geo-location represented by a tuple $\langle latitude, longitude \rangle$.

Due to different technologies and social aspects, personal trajectory data has also been logged by passive and active recordings (ZHENG, 2015). Passive recordings are generated, for example, by mobile phones automatically capturing the user's location. Complementary, active recordings are generated with users' intention, for example, by doing a check-in in a Location-based Social Network (LBSN), like Foursquare[4].

Data mining techniques have been applied to huge volumes of trajectory data, giving new dimensions to tasks like object tracking, urban planning, traffic management, market campaigns, and animal migration analysis (NOGUEIRA et al., 2018). These techniques can be combined with multiple applications, such as friendship suggestion (YANG et al., 2020), next place prediction (LIU et al., 2016), location-based advertisement (ALBANNA et al., 2015), and movement behavior analysis (FENG; ZHU, 2016).

An extensive collection of trajectory data mining approaches has been discussed to process, represent, and learn from it. The first research efforts focused on dealing with raw trajectory data. However, soon it became clear that the semantic gap between spatiotemporal data and contextual knowledge needed to be carefully investigated (SPACCAPIETRA et al., 2008).

Consider, for example, users' trajectories captured during many days in a big city and the task of uncovering their daily mobility patterns. Raw positional data is not enough to capture human' behaviors, like home-to-work and work-to-home patterns during workdays.

The semantics used to annotate the trajectory can significantly vary, depending on the data source, moving object, and application. A location point can usually be associated with a real-world place, such as an *Hotel*, or, more specifically, an *Ibis Hotel* (BOGORNY et al., 2009). Trajectory data from location-based networks may also contain textual information describing the users' feelings about the visited place. When considering vehicle trajectories, the constrained topological network where they move can be taken into account (FENG; ZHU, 2016).

According to Camossi et al. (2013), Semantic Trajectory has been mainly explored by three research areas: Spatiotemporal Data Modeling, for the representation of semantic trajectories (ATLURI et al., 2018); Knowledge Discovery from Data (KDD), for semantic trajectory mining (FENG; ZHU, 2016); and Geographic Visualization and Visual Analytics, for semantic trajectory visualization (BOGORNY et al., 2011).

Many surveys have been conducted in the literature as the semantic trajectory community evolved. Examples such as Parent et al. (2013), Albanna et al. (2015), Zheng (2015) and Feng and Zhu (2016) usually propose and describe their own trajectory data mining framework. Nonetheless, their base components are preprocessing, data management, query processing, trajectory data mining tasks, and applications.

[4] <https://foursquare.com>

Albanna et al. (2015) surveys the research on semantic trajectories and proposes a classification of study areas in three categories. This classification also relates to the data mining steps of representing, processing, and using data:

- Modeling: this area is interested in defining which part of trajectory data will be stored, how it will be stored and accessed, and which semantics will be used to annotate data;
- Computation: this area is interested in acquiring, processing, and annotating data in the suitable format defined by modeling;
- Application: once data has been preprocessed and represented, different tasks and applications can be explored.

In a positional paper, Laube (2015) reflects on the evolution of Geographical Information Science (GIScience) and spatial computing since the sudden data availability in the last two decades. It is argued that many tasks have been deeply explored and consolidated, including preprocessing, integrating, storing, managing, querying, and mining spatiotemporal data. On the other hand, other tasks still require much progress. Laube (2015) considers that it is necessary to embed context to trajectories in order to understand not only structural characteristics but the movement behaviors. Complementary, the integrated semantics needs to ideally have a similar spatial and temporal resolution to the original data.

While some studies focus on questions like *"what the objects move for"*, some other tries to analyze *"how they move"*, i.e., how to detect the objects' movement semantics (ALBANNA et al., 2015). Renso et al. (2013) argues that semantic is required to enable "business actionable knowledge" from trajectory data.

Based on these observations, more recent works have focused on combining different data sources to enrich trajectory data, like from Event-based Social Networks (e.g., Meetup[5], Eventbrite[6]), Location-based Social Networks (e.g., Foursquare, Twitter[7]), and from geo-databases as DBPedia and LinkedGeoData (FILETO et al., 2015b).

This dissertation is interested in abstract representations and already proposed frameworks that encapsulate data management steps. The subsection 2.3.1 discusses such representations and frameworks, and subsection 2.3.2 discusses Semantic Trajectory Data Mining, especially the pattern mining task. Finally, some publicly available datasets and their general characteristics are presented in subsection 2.3.3.

[5] <https://meetup.com>
[6] <https://eventbrite.com>
[7] <https://twitter.com>

2.3.1 Modeling and computation

In order to further enrich the discussion on Semantic Trajectory Modeling and Computation proposed in Albanna et al. (2015), we show in Figure 2.1 the data mining framework proposed in Feng and Zhu (2016). As shown in the image, there are multiple steps required to process data and retrieve useful information.

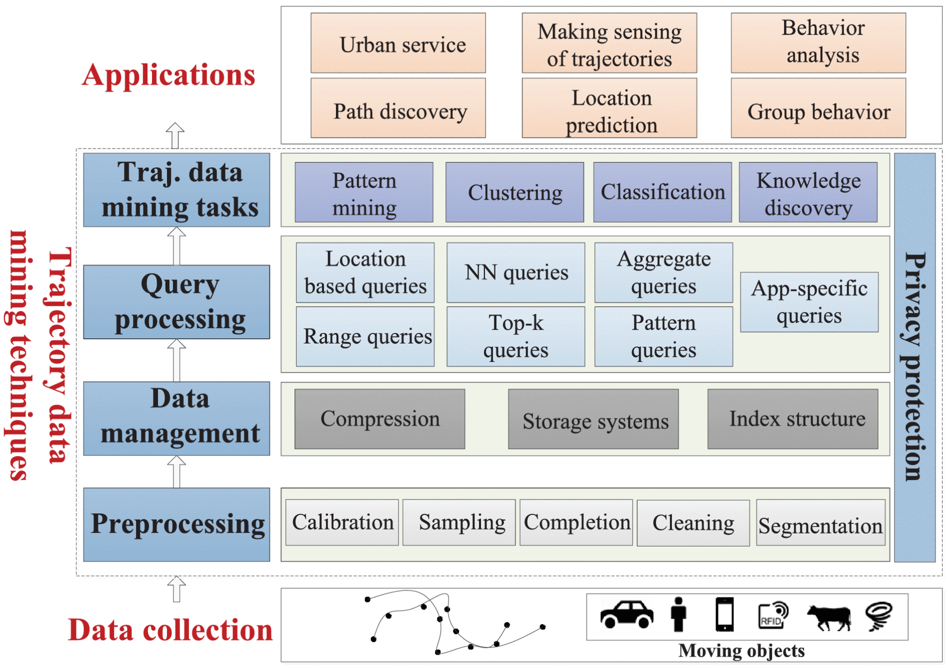


Figure 2.1 – A framework for Trajectory Data Mining. Source: Feng and Zhu (2016).

Given a set of raw trajectories, preprocessing tasks include noise filtering, sampling, stay point detection, trajectory segmentation, and map-matching (ZHENG, 2015; FENG; ZHU, 2016). Multiple raw data features can be used during a semantic enrichment step. For example, speed and direction can be used to infer semantics about the object's movement, e.g., transportation means (YAN et al., 2013), and time and spatial thresholds can be used with clustering algorithms to detect stops and be further cross related to Point-of-Interests (POIs) dataset.

The preprocessed trajectories need then to be stored in a suitable format to allow different queries over such data (FENG; ZHU, 2016). For example, a *location-based query* finds trajectories that are close to a set of queried locations, and a *range query* uses user-defined ranges in time and space to retrieve similar trajectories. Queries can be further generalized to retrieve similar trajectories based on similarity measures that consider multiple semantic dimensions (PETRY et al., 2019a).

Trajectory data may be stored and represented in GIS (Geographical Information Systems) databases, using relational schema to store and query geographical entities, trajectory data, and associated semantics. Spaccapietra et al. (2008) firstly investigated the requirements of trajectory data processing and data representation in the context of the European Project

GEOPKDD[8] (BOGORNY et al., 2014). They firstly described a trajectory as a sequence of *stops* and *moves*.

Stops represent important physical places for which the moving object manifested some interest, for example, by staying in that region for some time. As the object goes from one stop to the next, the movement characteristics are captured by the *move* concept. Stops and moves can be generated from processing raw data, allowing a more abstract and semantic representation of trajectories.

This first conceptual view of trajectories proposed in Spaccapietra et al. (2008) further inspired many works that extended it to represent different semantic concepts, like in Parent et al. (2013) and Bogorny et al. (2014), or to use alternative storages, like ontologies (NOGUEIRA et al., 2018).

In the context of trajectory data, ontologies can be used to represent entities and relationships related to space, time, and semantics (ALBANNA et al., 2015). An ontology representation allows us to integrate and interlink data from different sources (FILETO et al., 2015b), apply reasoning to infer new knowledge (NOGUEIRA et al., 2018) and to adhere to the Semantic Web standards (MELLO et al., 2019).

Different works propose and use ontologies. For example, Baglioni et al. (2009) represents Spaccapietra et al. (2008)'s stop-move as an ontology. They use DL axioms to state movement behavior patterns, allowing them to use domain knowledge and ontology reasoning to infer behaviors.

Bogorny et al. (2010) continues the discussion and argues in favor of considering the data mining process during the database schema design. They extend the stop-move representation in order to support the storage of mined patterns.

Bogorny et al. (2014) presents a conceptual data model called CONSTAnT (CONceptual model of Semantic TrAjecTories) extending Spaccapietra et al. (2008) and other previous trajectory representations. CONSTAnT aims to represent different semantic concepts, allowing the user to store semantic data in each trajectory point. CONSTAnT introduces the concept of semantic subtrajectories, for which semantics can be specified. It also allows modeling behaviors, goals, environment conditions and events happening in a given place.

Fileto et al. (2015b) proposes *Baquara*^{2}, a conceptual framework for semantically enriching and analyzing trajectory data, and a domain-independent ontology used to model data. *Baquara*^{2} allows the representation of different trajectory data granularities and considers the usage of LOD databases such as DBPedia and LinkedGeoData to enrich geo-located data further. Two main steps are defined: data preprocessing and data linking. Datasets from Twitter and Flicker are used to demonstrate the framework.

[8] Geographic Privacy-Aware Knowledge Discovery and Delivery: <http://infolab.cs.unipi.gr/projects/GeoPKDD/>

Nogueira et al. (2018) also propose a framework for semantically enriching and querying trajectory data, supported by the proposed STEP[9] ontology. Different granularities can be expressed, and rules are used to infer facts during queries. The FrameSTEP framework is proposed together with the STEP ontology, serving as a bridge between the STEP ontology and annotation algorithms in an object-oriented implementation.

The framework is also composed of a collection of utilities that compute simple attributes for each raw point (e.g., speed and time duration) and allows interfacing with different trajectory algorithms. The object-oriented entities can be exported as a Semantic Web compatible representation.

In STEP, a moving object (`Agent`) is associated with multiple trajectories, and each `Trajectory` can be associated with `Semantic Description`'s through a `Feature Of Interest` (`FOI`) instance. `Semantic Description`'s can be quantitative or qualitative, and are encouraged to be extended to specific use cases.

Further details about the STEP ontology are postponed to Section 3.3, where they are discussed in this work's context. A particular focus in distributing the STEP ontology online is taken, aligning it with the Semantic Web and LOD ideas.

Finally, the recently proposed MASTER model (MELLO et al., 2019) explores the definition of a trajectory data representation that can represent arbitrarily complex semantics while also exploring the logical and physical technologies involved in querying trajectory data. The logical model is implemented using RDF.

It focuses on being simple yet presenting a powerful expressiveness. Semantic aspects can be of any type of data and can be associated with multiple trajectory granularity levels, including a whole trajectory or a single point, and including the moving object itself.

In summary, we can consider that the trajectory ontologies proposed in the literature are evolving to be simpler, allow multiple granularities, and allow generic yet compact ways to store multiple semantic aspects about any trajectory-related concept.

2.3.2 Data mining and Association Rules

Once preprocessing algorithms have been applied to raw trajectories, and a suitable representation has been populated with trajectories and semantics, we can use different data mining tasks to serve many applications. Next, we briefly review such tasks and then focus on Trajectory Association Rules.

Trajectory Data Mining tasks include classification, anomaly detection, and pattern mining (FENG; ZHU, 2016; ZHENG, 2015). The last can be further divided into different

[9] <http://purl.org/net/step>

patterns, such as group patterns, frequent patterns, sequential patterns, and association rules. Trajectory clustering can also be considered a pattern mining problem (ZHENG, 2015).

Trajectory classification aims to associate labels to trajectories or trajectories segments. Such labels can relate, for example, to transportation modes or human activity. Sequence inference models such as Dynamic Bayesian Networks (DBN), Hidden Markov Models (HMM), and Conditional Random Field (CRF) are usually used to condition labels to local and adjacent trajectory points. Bogorny et al. (2014) discuss different methods that can be applied to infer semantics from the raw trajectory. The extracted semantic data may be further used to improve other pattern mining tasks.

Regarding *trajectory anomaly detection*, Zheng (2015) divides it into two groups: outlier trajectories and anomalous events. The first one uses distance metrics to detect trajectories that are significantly different from others. Clustering and pattern mining can be used to detect such outliers. In the case of anomalous events, they can be identified by considering many trajectories, such as when detecting traffic anomalies caused by accidents or protests.

Trajectory patterns can be considered spatiotemporal evidence of movement behavior, which in turn is a complex process and depends on many contextual variables (BAGLIONI et al., 2009). The representation chosen to represent trajectories limits the kind of patterns that can be extracted. Nonetheless, semantic patterns are independent of the geolocations, usually being sparse in time and having no geometric similarity (BOGORNY et al., 2009).

As previously listed, there are many different pattern categories. For example, group patterns are subsets of trajectories that move together for a specific time, such as a flock or a swarm. To detect such groups, one may cluster trajectories by using distance metrics that consider trajectory components such as spatial dispersion, temporal duration, movement velocity, and heading direction (ZHENG, 2015; FENG; ZHU, 2016). Similarity measures can also be defined to consider any type of trajectory attribute and semantics (PETRY et al., 2019a).

Finally, many different approaches have investigated frequent patterns, sequential patterns, and association rules on spatial and spatiotemporal data (HUANG et al., 2004; VERHEIN; CHAWLA, 2008; MONREALE et al., 2009; SENGSTOCK; GERTZ, 2013). In the context of trajectory data, mined rules can communicate mobility patterns and may be used for reasoning or to predict knowledge such as a user's next place.

We define three contexts in which Trajectory Data Mining literature has explored the usage of rules:

- *Mining Spatiotemporal rules*: most of the works on rule mining focus on extracting patterns from raw trajectories. These approaches usually divide space in a grid and apply transaction-based strategies to mine patterns relating associations of visiting different grid spaces at different times (HUANG et al., 2004; VERHEIN; CHAWLA, 2008; MONREALE et al., 2009; SENGSTOCK; GERTZ, 2013).

- *Handcrafted rules*: domain experts may use rules to capture domain knowledge. These rules can then be used, for example, to apply data constraints, to infer new facts (BAGLIONI et al., 2009; RENSO et al., 2013), and to retrieve interesting entities (CAMOSSI et al., 2013). These usages are usually associated with ontology-based representations since they offer languages for expressing complex rules.

- *Mining Semantic Trajectory rules*: Some works incorporate semantics aspects while also automatically mining rules. Semantic Trajectory data is considered as a transaction-based dataset and Apriori or similar variations are used (BOGORNY et al., 2009; RIZK; ELRAGAL, 2012; MOUSAVI et al., 2016; KHOSHAHVAL et al., 2017; ZHANG et al., 2019)

Next, we review these works based on the classification introduced.

Mining Spatiotemporal rules

Huang et al. (2004) study the task of detecting colocation patterns in spatial data. A colocation pattern is a set of objects or event classes that often occur in close geographic proximity (SENGSTOCK; GERTZ, 2013). In this case, the objects and events are defined as Boolean spatial features, indicating, for example, a species' occurrence at a given geographic area. Accordingly, a colocation rule can be defined as an association rule between these spatial features.

A colocation pattern can express, for example, symbiotic species, and then a colocation rule can be used to represent the association between the two species co-occurrence (HUANG et al., 2004). The mining algorithm relies on a user-specified relation that specifies whether two geographic instances are neighbors. Based on this relation, the authors define a *transaction* and then apply a transaction-based mining algorithm similar to Apriori.

Verhein and Chawla (2008) define Spatio-Temporal Association Rules (STARs), which are association rules that describe how objects move between regions over time. Given a set of objects moving throughout a fixed set of regions, the mined STARs can predict how objects will move between the regions. Experiments with real animal tracking data show that STARs can get insights into individuals' and groups' movement patterns.

As argued in Verhein and Chawla (2008), STARs cannot be mined by transaction-based algorithms. Also, specific spatial-aware metrics are defined to consider, for example, the different area sizes of the regions. Nonetheless, STARs do not consider semantic aspects and have a very restricted format, describing only an object's sequential appearance in two different regions.

Monreale et al. (2009) generate association rules from T-patterns (GIANNOTTI et al., 2007) which are a generalization of sequential patterns with temporal distance thresholds. These rules are used to predict a user's next location. Special care is taken to vary the temporal threshold and the minimum support to maximize the mined rules' prediction power. The authors define metrics to assess the rules' spatio-temporal coverage, dataset coverage, and spatial size since these aspects influence the usefulness of a set of rules in the prediction task at hand.

Sengstock and Gertz (2013) extend the Frequent Itemset Mining framework to deal with geo-transactions. In this context, a geo-transaction is a transaction with an associated spatial point, i.e., it can have items relating to geolocations. The definition of an itemset and the associated metrics are extended to include spatial characteristics. To achieve that, the authors define a spatial density function of the points associated with the items in the itemset.

The task of spatial itemset mining is defined on a geo-transaction database, where each mined itemset has spatial characteristics that can be used to understand the data better. A dataset from the Flickr LBSN is used, and each transaction is a combination of geolocation and a set of tags associated with the Flicker's post. The experiments show how tags co-occur on different spatial areas' granularities.

Handcrafted Rules

Baglioni et al. (2009) proposes the Athena architecture to represent the steps of semantic enrichment, ontology representation, and querying. The framework allows the user to specify inference rules to an ontology reasoner and then query the inferred facts. Real GPS data is used to analyze Milan's tourist movements, and a reasoner is used to infer facts such as *TouristActivity*, defined by handcrafted patterns. The inferred facts are then used to group similar trajectories with clustering trajectory algorithms. The experiment shows two distinct tourist behaviors: tourists coming from outside the city and moving to the center, and tourists moving from the center to outside.

Renso et al. (2013) continues the development of Athena, and two new applications are provided. The first one uses vehicle GPS trajectories to detect the *Home-Work* behavior, defined as a DL axiom that considers the trajectories' origins and destinations. The second application uses a dataset of GPS readings of visitors of the *Dwingelderveld National Park*, in the Netherlands. Visitor behaviors are also defined as axioms and then inferred using the ontology reasoning capabilities.

Another example of rules built by domain experts is presented by Camossi et al. (2013), where they are used to detect anomalous events. Episodes related to container transportation are represented in a stop-move ontology. Each stop is semantically enriched with container information and vessel events. The formal representation allows the authors to use DL axioms to reason on trajectory data automatically. They test the representation with a dataset containing millions of events and show that DL queries' expressivity can be used to describe complex anomalous behaviors. Using the reasoning capabilities of the ontology-based representation, they retrieve suspicious/anomalous container trajectories.

Mining Semantic Trajectory rules

Bogorny et al. (2009) propose the ST-DMQL query language and explore mining frequent patterns, sequential patterns, and association rules. They divide each of these tasks into move patterns and stop patterns. This means that patterns must consider either only stops or only moves episodes. A trajectory item is then defined to extend the concept of an item from pattern mining by combining information about space and time of stops/moves.

The ST-DMQL allows queries to retrieve patterns and association rules, and includes features such as time granularity (e.g., 8 am, 8 am-11 am, morning) and stop category hierarchy (e.g., museum, monument, tourist place). Although the underlying mining algorithm can be changed, the patterns mined in ST-DMQL assumes the existence of the definition of a traditional item and itemset from transactional pattern mining (Fournier-Viger et al., 2017).

Rizk and Elragal (2012) propose a framework for raw trajectory preprocessing and semantic enrichment. Although an ontology-based representation is used, a propositional view transformation is applied before executing a traditional association rule mining algorithm. A prototype using synthetic data was validated by one public and one private stakeholder of the *Tourism and Travel* domain. The mined rules were considered useful for decision making, but real data would be required to use the mined knowledge. The stakeholders made two improvement requests: a visual component to display the results, and the integration with an events repository, inserting more context to the trajectories and consequently to the mined rules.

Mousavi et al. (2016) argument that ontologies may have a crucial role in the knowledge discovery process on semantic trajectories, especially when mining association rules. They model an ontology-based on the stop-move representation. A stop is associated with a place and an activity, and a move is associated with a transportation mean. The Apriori algorithm is applied to a propositional view of the ontology trajectory. Mined rules are applied to location-based services to trigger advertisements based on spatiotemporal context.

Ghosh and Ghosh (2018) mine patterns using a pre-defined set of rule templates. They argue that some rules might be interesting when considering a specific user's behavior or considering a subset of users (e.g., a group of users). Therefore, they propose two different interestingness metrics to account for interesting patterns that are true for subsets of users.

After applying a semantic enrichment process, Zhang et al. (2019) apply the PrefixSpan algorithm to mine sequential trajectory patterns. They consider a trajectory as a sequence of stops annotated with a venue category. They can mine patterns such as "Apartment $\rightarrow$ Subway Station $\rightarrow$ Office $\rightarrow$ Mall". They also observe that a fixed support threshold might be inconvenient since it discards rules that might hold for an interesting data subset.

In summary, Semantic Trajectory Pattern Mining has three main tasks: frequent patterns, sequential patterns, and association rules mining. Table 2.1 summarizes the main works discussed in this section, including the different tasks and algorithms used. Some works use querying languages to query relational representations of semantic trajectories. In this case, sequences of episodes are usually retrieved based on properties such as place category or time.

When an ontology-based representation is used, the domain-specific relations are added by handcrafted rules to serve querying tasks. When mining algorithms are used, they are not defined as to explore relations between trajectory episodes, other than their explicit sequence and individual properties.

For example, consider a relation that connects two events if they happen within a two-hour time window. Traditional trajectory rules mining would not be able to mine patterns considering this relation. Our proposed approach is to use mining algorithms that can explore such complex relationships. The ontology representation can be extended using domain knowledge to incorporate relations between episodes and their attributes, allowing new associations to be mined. Combining external data sources and knowledge bases as background knowledge further increases the semantic richness used to describe trajectory patterns.

As we mine these more complex rules, we also increase the number of uninteresting rules that may be mined. Approaches to better filter these rules will need to be investigated based on what kind of patterns are mined. For example, rules combining knowledge acquired from different data sources may be considered to be more interesting than rules that use only one data source since they capture patterns found by analyzing a broader trajectory context.

Table 2.1 – Comparative table of tasks and algorithms for trajectory pattern mining

Paper	**Task**	**Algorithm**	**Data input representation**	**Metrics**
Huang et al. (2004)	Colocation rules mining	Custom	Not specified	Confidence
Verhein and Chawla (2008)	Spatio-temporal rules mining	STAR-Miner (Custom)	Not specified	Support/Confidence
ST-DMQL (BOGORNY et al., 2009)	Association Rule Mining	Any transaction-based algorithm (with implemented interface)	Propositional	Support/Confidence
Camossi et al. (2013)	Reasoning using DL rules	Any ontology reasoner	Ontology	-
Sengstock and Gertz (2013)	Spatial Itemset Mining	Based on FP-growth	Propositional + Spatial Density function	Spatial Support/Confidence
Athena Baglioni et al. (2009), Renso et al. (2013)	Frequent Pattern Mining	Agrawal et al. (1993)	Propositional	Support/Confidence
Rizk and Elragal (2012)	Association Rule Mining	FP-growth	Propositional	Support/Confidence
Mousavi et al. (2016)	Association Rule Mining	Apriori	Propositional	Support/Confidence

2.3.3 Data and datasets

As previously discussed, different technologies can be used to collect trajectory data. Also, the research community has been publicly providing many datasets used in different tasks and applications. Zheng (2015) discusses some of the primary datasets and related researches. However, most of the trajectory data contain only raw readings.

All data (except synthetic data) can be semantically enriched through preprocessing or data integration from external data sources, such as the Google Places API[10], Foursquare API[11], and the LinkedGeoData database. Nonetheless, data integration has many problems, such as data unavailability, data sparsity, and different time and place resolutions.

Fileto et al. (2015a) characterizes two important concepts: *raw trajectory* and *user trail*. As defined by them, a raw trajectory is a time-sorted sequence of spatiotemporal positions occupied by an object. A user trail is a time-sorted sequence of geo-referenced registries of user interactions with a particular system. This system can be, for example, Location-based Social Networks, such as Twitter, Foursquare, and Flickr.

Raw trajectory data can be obtained using very precise sensors and sampled with a high frequency. On the other hand, user trails are usually sparse, biased to a specific set of geolocations, and lagging in time. Nonetheless, while it is hard to acquire annotated raw trajectory data, user trails usually have associated text, images, and other semantics. Raw trajectory and user trail can be fused as in Gil et al. (2014) and Fileto et al. (2015b).

GPS-data

GPS-enabled devices allow capturing trajectory data of people, animals, and vehicles (such as taxis (YUAN et al., 2010), trucks and motor-homes (SENOZETNIK et al., 2019)). Usually, semantic data is not part of the data collection. An exception is when users are instructed to describe their activities as in projects such as GeoLife Project (ZHENG et al., 2010) and TagMyDay (PETRY et al., 2019a).

Synthetic

Although raw trajectory data is plenty available, semantic trajectory data is not. This means that preprocessing transformations must be applied to raw trajectory data in order to annotate it semantically. Also, ground-truth annotations are usually not available.

Pelekis et al. (2016) approach this problem by building Hermoupolis, described as a pattern- and semantic-aware synthetic semantic trajectory simulator. Given a set of mobility profiles, Hermoupolis produces semantically annotated trajectories. The simulation is further improved by allowing real trajectory data to serve as the mobility profiles. This can be viewed

[10] <https://developers.google.com/places/web-service>
[11] <https://developer.foursquare.com>

as a data augmentation processing that uses profiles from small real datasets to generate new trajectories.

Hermoupolis generates network constrained trajectories, which means that trajectories are related to a road network. Each stop and move can have associated tags indicating, for example, a POI's category or a transportation mean.

Zheng (2015) and Pelekis et al. (2016) discuss other trajectory simulators.

Location-based Social Networks

People's physical and digital life intersect as users of social media feed their geo-position in social networks. This data category has different characteristics when compared to GPS data. They are sparse in time and space, and there is a bias for some categories of places.

Datasets captured from location-based social networks exists, such as from Gowalla (CHO et al., 2011), Brightkite (CHO et al., 2011), Foursquare (Dingqi Yang et al., 2015), and Flickr[12] (TAKIMOTO et al., 2017). Raw and Semantic Trajectories can be combined, such as in Gil et al. (2014) where GPS data and geo-tagged tweets are fused into a single representation.

Others

RFID data is another trajectory data source, but less explored in literature (WANG et al., 2012). Also, datasets about hurricanes trajectories exist, such as the Hurricane Trajectory dataset (HURDAT2) (LANDSEA; FRANKLIN, 2013), provided by the National Hurricane Service (NHS). Containers trajectory is used in Camossi et al. (2013).

2.4 Rule learning on Knowledge Bases

As discussed in Section 2.1, there are many public and private Knowledge Bases. Many of them are built using a semi-supervised approach by structuring data from different data sources (SUCHANEK et al., 2019). Excluding carefully curated knowledge bases, the KBs' facts are limited and noisy.

For example, although a KB intends to model the real world or a part of it, the data sources used to populate it with facts usually do not contain all knowledge expected to be stored. This implies that this KB is not fully complete. Complementary, data sources may contain wrong knowledge, or wrong knowledge may be generated during the knowledge extraction step. This means that the KB will not be entirely correct (PAULHEIM, 2016; SUCHANEK et al., 2019).

To overcome these issues, researchers in the field of *Knowledge Base Refinement* have proposed many methods that can improve a Knowledge Base's coverage and correctness. Paul-

[12] <https://flickr.com>

heim (2016) defines three orthogonal approaches to refine a KB. Firstly, the goal of the method may be to increase KB's completion or KB's correctness. Secondly, the methods may focus on specific targets, considering the refinement of only entities or only relationships, or even a subset of them. Finally, the refinement methods may have only the KB itself as input, or use additional external data.

One approach to the KB refinement problem is mining association rules. As discussed in Section 2.2, in the context of relational data, association rules can be used to describe general regularities that hold in a database. In KBs, association rules can also be used to infer new facts and check the consistency of the existing ones (GALÁRRAGA et al., 2013).

In Lajus et al. (2020), the authors divide the KB rule mining literature into two generations. The first generation includes algorithms such as WARMR (GOETHALS; Van den Bussche, 2002), which are based in Inductive Logic Programming (ILP). These algorithms have two problems when applied to current KBs: they do not scale to huge volumes of data, and they require negative facts as input.

These are significant issues since current KBs contain millions of facts, and they usually do not contain negative statements. This led to the development of the second generation of algorithms. The first one was AMIE (GALÁRRAGA et al., 2013), designed with properties tailored to working in the KB refinement context.

In subsection 2.4.1, we talk about *Multi-relational Association Rules*, a generalization of association rules intended to work on relational databases. In subsection 2.4.2, we present AMIE, a well-known state-of-the-art algorithm for KB rule learning. We also take the opportunity to discuss in subsection 2.4.3 about a series of works that have not been previously related to the KB rule learning literature but may contribute to future discussions. This set of works was the original inspiration for the proposed investigation of the AMIE algorithm as a general data mining tool. Finally, subsection 2.4.4 briefly review multiple algorithms that mine logical rules from RDF data.

2.4.1 Multi-relational Association Rules

This section talks about *Multi-relational Association Rules*, as defined in Dehaspe and Raedt (1997) and Džeroski (2009). We also talk about WARMR (DEHASPE; RAEDT, 1997), an important algorithm for mining multi-relation rules.

Multi-relational Data Mining (MRDM), some times called only Relational Data Mining (RDM), is a subfield of data mining concerned with mining knowledge from multiple tables (relations) of a relational database (DŽEROSKI, 2009). This contrasts with typical data mining algorithms, which require all potential data to be aggregated or joined in a single table.

In traditional data mining, patterns are mined from this single table and represented in *propositional logic*. This approach is, therefore, called attribute-value or propositional learning.

In Multi-relational Data Mining, patterns are represented as a subset of *first-order logic*, and its approaches are called first-order learning or relational learning.

The basic concepts in first-order logic are *predicates* (e.g. $marriedTo$) and *variables* (e.g. x, y). Variables are usually represented in lowercase, while uppercase names represent constants, i.e., concrete entities. These concepts can be combined to state that a person X is married to a person Y through the fact $marriedTo(X, Y)$.

A relation in a relational database is equivalent to a predicate in first-order logic (DŽEROSKI, 2009). In traditional association rules mining, as defined in Chapter 2 in the market basket domain, we can consider that the implicit relation *boughtTogether* connects all items in a transaction. In multi-relational data mining, the relations are explicitly represented, and multiple relations can be combined in the same rule (RAMEZANI et al., 2014).

Instead of using MRDM specific approaches, one can create a single table combining multiple tables from a relational database in a process called propositionalization. This approach's problems are efficiency concerns and limited expressiveness since propositionalization may not capture the same original data semantics. Džeroski (2009) discusses that considering data in multiple tables allows overcoming the simplification required during joining or aggregating data in a propositional representation.

Considering this dissertation's context, we focus on general aspects of Multi-relational Data Mining and Multi-relational Association Rules. For a complete discussion on MRDM, including the multi-relational version of standard algorithms such as Decision Trees, we refer the reader to Džeroski (2009).

Multi-relational Data Mining is mainly based on Inductive Logic Programming (ILP), which is at the intersection of *machine learning* and *logic programming*. Logic programming is concerned with *deductive* inference given a set of first-order facts. On the other hand, Inductive Logic Programming is interested in *inductive* inference.

The most common task in ILP is to learn logical definitions of relations. Instances that belong or do not belong to a target relation are used as examples to generate hypotheses about unseen instances. The hypotheses are expressed as rules that use relations given as background knowledge to infer the target relation (DŽEROSKI, 2009).

Just as in traditional data mining, MRDM algorithms search the space of patterns using approaches that minimize the overall algorithmic complexity. Patterns are iteratively created by applying refinement operators on previously found patterns. Traditional frequent itemset mining has refinement operators that can add a new item to an item set. In MRDM, the refinement operators include adding a new relationship or variable (DŽEROSKI, 2009).

To make the data mining task feasible, it is usually required to explicitly provide constraints to the algorithms to limit the search space. Some constraints include specifying what relations should be used in patterns and how the relations can be interconnected. The explicit specification of constraints applied to the search space is called *declarative bias* (DŽEROSKI, 2009).

WARMR is a multi-relational association rule mining algorithm proposed in Dehaspe and Raedt (1997). It is a modification of Apriori that generalizes the concept of an *itemset* to the concept of an *atomset*, i.e., a set of logical atoms where each atom is a tuple in a relational database table. The algorithm has as input a relational database, a *minfreq* threshold, and a declarative language bias used to express the declarative bias desired.

WARMR upgrades two characteristics of Apriori. Firstly, the definition of support is given by the number of answer substitutions of the mined pattern. Secondly, the candidate query generation is defined in order to account for the more complex refinement operators and to the declarative bias (DŽEROSKI, 2009).

WARMR was then extended in Goethals and Van den Bussche (2002), as WARMeR. It is a modified approach that supports a broader range of conjunctive queries and increase efficiency of search space exploration (GALÁRRAGA et al., 2013).

2.4.2 AMIE

Galárraga et al. (2013) investigated the task of rule mining in the context of Knowledge Base Refinement. They propose AMIE, a multi-threaded, heavily memory-indexed algorithm to mine Horn Rules from an RDF triples dataset. AMIE can mine rules such as $hasChild(m, c) \wedge marriedTo(m, f) \rightarrow hasChild(f, c)$, which can be read, given an associated probability, as “if two people are married and one of them has a child, then the other person has the same child”.

The algorithm can be applied to any RDF-graph, making it suitable for working on the Semantic Web KBs, such as YAGO and DBpedia. As discussed in Galárraga et al. (2013), their approach is related to traditional Association Rule Mining and Logical Rule Mining, i.e., ILP systems and Multi-relational Data Mining.

The algorithm implementation was improved by an optimized version, AMIE+ (GALÁRRAGA et al., 2015), and then again by AMIE 3 (LAJUS et al., 2020). Both of them implement runtime and memory optimizations based both on implementation decisions and mining strategies.

In the original paper, Galárraga et al. (2013), AMIE is compared with WARMR and ALEPH, which are both ILP systems. The experiments reported show that these two approaches mine less interesting rules and take much longer than AMIE. Ontological Pathfinding (CHEN et al., 2016) and RudiK (ORTONA et al., 2018) are AMIE+ alternatives proposed to mine rules faster. Lajus et al. (2020) shows that AMIE 3 is more general and faster than those approaches.

Next, we discuss the original AMIE’s main properties since AMIE+ and AMIE 3 introduced speed-ups without changing the underlying mining approach. Similar to Multi-relational Association Rules, AMIE can mine rules involving predicates. We base our descriptions and notations on Lajus et al. (2020) and Petri and Silva (2020). We present the following initial definitions:

Knowledge Base: A knowledge base $\mathcal{K}$ is a set of facts $r(s, o)$, where the subject s belongs to a set of entities $\mathcal{I}$, the relation r belongs to a set of relations $\mathcal{R}$, and the object o belongs to $\mathcal{I}$ or a set of literal values.

Relations and Functions: Given a relation r and the set of facts $r(s, o) \in \mathcal{K}$, the inverse relation of r, denoted r^-, consists of all the facts $r^-(o, s)$. A relation r is a function in $\mathcal{K}$ if r has at most one object for each subject. The notion of functions has been generalized to the *functionality score* of a relation r (SUCHANEK et al., 2011):

$$fun(r) = \frac{|\{s : \exists o : r(s, o) \in \mathcal{K}\}|}{|\{(s, o) : r(s, o) \in \mathcal{K})\}|} \tag{2.1}$$

The functionality score is 1 for strict functions, close to 1 for quasi-functions, and it is smaller for relations that have many objects for each subject. If $fun(r) < fun(r^-)$, AMIE implicitly uses r^- during mining. By doing so, we can intuitively say that a fact $r(s, o)$ is a fact about s, or an r-attribute of s (GALÁRRAGA et al., 2013).

Atoms and Rules: An *atom* is an expression of the form $r(X, Y)$, where r is a relation and X, Y are either constants or variables. We denote variables by lowercase letters, whereas constants (entities or literals) are always capitalized. An atom is *instantiated* if at least one of its arguments is a constant. A (conjunctive) *query* is a conjunction of atoms $\mathbf{B}$, which we separate by commas: $B_1, \ldots, B_n$. A *substitution* σ, which can be applied to an atom or a query, is a partial mapping from variables to constants. A (Horn) *rule* is a formula of the form $\mathbf{B} \Rightarrow H$, where $\mathbf{B}$ is a query of body atoms, and H is the head atom.

For completeness, Galárraga et al. (2013) try to establish some parallels to Association Rule Mining. Consider that we are interested in mining rules relating exactly n entities. We can create one transaction for every set of n entities that are connected to the KB.

For example, Table 2.2 shows three transactions identified by the C_i entities that appear in the transaction, where $1 \leq i, j \leq n$. A transaction identified by $\langle C_1, \ldots, C_n \rangle$ contains a relation $r(x_i, y_j)$ if $r(C_i, C_j)$ is in the KB. In this representation, each relation is an item, and a traditional association rule mining algorithm can be applied. A mined rule is in this case a Horn rule. For example, consider that Apriori mined the association rule $\{marriedTo(x_1, x_3),\ hasChild(x_3, x_2), \} \rightarrow \{hasChild(x_1, x_2)\}$. This rule is equivalent to the Horn rule $marriedTo(x_1, x_3),\ hasChild(x_3, x_2) \rightarrow hasChild(x_1, x_2)$.

Representing the entire KB as a table of transactions is infeasible as the number of all possible combinations is prohibitively large. On the other hand, AMIE deals directly with the RDF-graph triples and uses many techniques to deal with the huge volume of data while still mining interesting and relational rules.

Table 2.2 – Approach for mining rules with 3 variables using traditional association rule mining. Source: Galárraga et al. (2013).

Transaction Label	**Transaction Items**
⟨Elvis, Lisa, Priscilla⟩	$\{hasChild(x_3, x_2), hasChild(x_1, x_2), marriedTo(x_1, x_3)\}$
⟨Barack, Mali, Michelle⟩	$\{hasChild(x_3, x_2), hasChild(x_1, x_2), marriedTo(x_1, x_3)\}$
⟨François, Flora, Ségo⟩	$\{hasChild(x_3, x_2), hasChild(x_1, x_2)\}$

Another example of a rule that can be mined by AMIE from the YAGO KB is $hasChild(p, c), isCitizenOf(p, s) \rightarrow isCitizenOf(c, s)$. This rule can be instantiated, which means that some variables were substituted by entities. If we can instantiate all atoms in the body using facts that appear in the KB, then the rule's instantiated head is a prediction. For example, the above rule can predict that Lisa is American ($isCitizenOf(Lisa, USA)$) if the KB contains the fact that a parent of Lisa ($hasChild(Elvis, Lisa)$) is American ($isCitizenOf(Elvis, USA)$) (GALÁRRAGA et al., 2013).

Formally, we may define new facts inference as follows:

Predictions: Given a rule $R = B_1, \ldots, B_n \Rightarrow H$, and a substitution σ, we call $\sigma(R)$ an *instantiation* of R. If $\sigma(B_i) \in \mathcal{K}\ \forall i \in \{1, \ldots, n\}$, we call $\sigma(H)$ a *prediction* of R from $\mathcal{K}$. We say that $\mathcal{K} \wedge R$ entails $\sigma(H)$ and write $\mathcal{K} \wedge R \models \sigma(H)$. If $\sigma(H) \in \mathcal{K}$, we call it a *true prediction*. If the prediction is not in the KB, then we must decide whether to consider it as a counter-example of the rule (a false prediction), or to assume it as a plausible prediction.

Since KBs do not store negative facts (see Section 2.1), we need to define strategies to generate the rule's counter-examples (SUCHANEK et al., 2019). KBs work under the Open-World Assumption (OWA), which means that any fact not in the KB cannot be assumed as false. On the other extreme, traditional databases (and association rules) work under the Closed-World Assumption (CWA), which means that any fact not in the database is considered to be false. This means that we have no counter-examples under OWA, and under CWA, we have no flexibility to predict new facts.

Galárraga et al. (2013) propose the *Partial Completeness Assumption* (PCA) to generate counter-examples in AMIE. It considers that if the KB knows some r-attribute of s, then it knows all r-attributes of s. This is mostly true for relations with high functionality (such as $hasBirthday$) and a reasonable assumption for many relations (such as $hasNationality$). If r^- has a grater functionality score than r, we may exchange $r(s, o)$ by $r^-(s, o)$.

AMIE defines a set of metrics used to prune, filter and evaluate the mined rules' interestingness. Those metrics are:

Support: The *support* of a rule $R = \mathbf{B} \Rightarrow r(x, y)$ in a KB $\mathcal{K}$ is the number of true predictions $r(X, Y) \in \mathcal{K}$ that the rule entails, as shown in Equation 2.2.

$$supp(\mathbf{B} \Rightarrow r(x, y)) = |\{\, r(x, y) : (\mathcal{K} \wedge R \models r(x, y)) \wedge r(x, y) \in \mathcal{K}\,\}| \tag{2.2}$$

This definition of support is interesting since it is guaranteed to decrease monotonically, i.e., adding new atoms to the rule's body does not increase the number of head ($r(x, y)$) instantiations. This monotonicity allows a pruning opportunity during patterns generation.

AMIE defines the proportional version of the absolute support, called *Head Coverage*. It is the ratio between support and the number of the relation's instantiations, as shown in Equation 2.3.

$$hc(\mathbf{B} \Rightarrow r(x, y)) = \frac{supp(\mathbf{B} \Rightarrow r(x, y))}{|\{\,(x', y') : r(x', y') \in \mathcal{K}\,\}|} \tag{2.3}$$

Confidence: The confidence of a rule R in a KB $\mathcal{K}$ is the proportion of *true predictions* out of the *true predictions* and *false predictions*. AMIE uses two confidence metrics, Standard-Confidence (*Std Conf*) and PCA-Confidence (*PCA Conf*). They are given by Equation 2.4, where *cex* denotes the counter-examples of R. In *Std Conf*, any predicted fact not in $\mathcal{K}$ is considered a counter-example. In the *PCA Conf*, only facts which contradicts the PCA are said to be counter-examples.

$$conf(R) = \frac{supp(R)}{supp(R) + |\{\,p : (\mathcal{K} \wedge R \models p) \wedge p \in cex(R)\,\}|} \tag{2.4}$$

Since the PCA Confidence allows modeling predictions as unknown facts, it is called a completeness-aware rule metric (TANON et al., 2017). Tanon et al. (2017) further argues that completeness-aware assumptions should consider the trade-off on estimating "the number of wrongly predicted facts in complete areas and the number of newly predicted facts in known incomplete areas."

As discussed in subsection 2.4.1, ILP systems usually use declarative bias to limit the pattern search space. A similar strategy is used by AMIE, which outputs only *connected* and *closed* rules. Two atoms are *connected* if they share a variable or an entity. A rule is said to be *connected* if every atom is connected transitively to every other atom of the rule. Complementary, a rule is said to be *closed* if every variable in the rule appears at least twice (GALÁRRAGA et al., 2013).

These biases are important since mining only connected rules filters out rules containing unrelated atoms, and mining only closed rules removes rules that predict only the existence of a fact. For example, it filters out $diedIn(x, y) \rightarrow \exists z : wasBornIn(x, z)$, since we are interested in rules that predict facts, such as $diedIn(x, y) \rightarrow wasBornIn(x, y)$ (GALÁRRAGA et al., 2013).

Recursive rules are allowed to be mined, which means that the head relation can appear in the body. Also, AMIE can operate under a mode in which it tries to instantiate the variables that appear in the rule (GALÁRRAGA et al., 2013).

AMIE can be summarized as in Algorithm 2.2. The algorithm starts with an empty set of rules and then adds new relations/atoms iteratively. The decreasing monocity of head coverage is used to prune the rules.

Algorithm 2.2 AMIE rule mining. Adapted from Galárraga et al. (2013).

```
1: procedure AMIE(KB $\mathcal{K}$)
2:     $q \leftarrow empty\ queue$
3:     Execute in parallel:
4:     while $not\ q.isEmpty()$ do
5:         $r \leftarrow q.dequeue()$
6:         if $r$ is closed $and$ $r$ is not pruned for output then
7:             Output $r$
8:         for all refinement operators $O$ do
9:             if $r' := O(r)$ is not pruned then
10:                $q.enqueue(r')$
```

A rule in AMIE is a sequence of atoms, where the first one is the head, and the others form the rule's body. For each candidate rule, AMIE applies one of three refinement operators:

1. Add dangling atom: a new atom is added with a fresh variable for one of its arguments. The other argument (variable or entity) occurs in some other atom of the rule;
2. Add instantiated atom: a new atom is added with an entity for one argument of its arguments. The other argument (variable or entity) occurs in some other atom of the rule;
3. Add closing atom: a new atom is added with both of its arguments (variable or entity) occurring in some other atom of the rules.

The benefits of AMIE are that (i) it does not need any parameter tuning (other than the mining thresholds) or user interaction, (ii) it can run over millions of facts on a short time period, (iii) it mines rules with greater potential to infer new facts or represent data characteristics.

AMIE has limited support for using the ontology schema. The mined rules may use the *rdf:type* relation to specifying the domain and range of the head relation. Also, rules involving numerical expressions cannot be mined. Some theoretical work for mining numeric rules has been discussed in Galárraga and Suchanek (2014), but no implementation has been done. AMIE 3 implementation and documentation is publicly available[13].

2.4.3 Description Logics rules

The STAR-CITY project (**S**emantic **T**raffic **A**nalytics and **R**easoning for **CITY**) was developed by IBM researchers and aimed in giving insights on historical and real-time city traffic conditions. These insights could be used by, for example, transportation departments, allowing better traffic management.

STAR-CITY system is fed with semantically annotated data generated by many different sources in different velocities. For example, the static city map is combined with a real-time bus

[13] <https://github.com/lajus/amie>

location data stream. The project was initially applied in Dublin City, Ireland (LÉCUÉ et al., 2014a), and then tested on many other cities with different requirements (LÉCUÉ et al., 2014c).

One important concept in the STAR-CITY system is ontology streams, which are required to represent the input data stream. An *ontology stream* is defined in Ren and Pan (2011), and can be viewed as a time-ordered sequence of snapshots of an ontology's A-Box. This means that the T-Box is the same for all snapshots, while the set of instances and their relationships are constantly updated with new pieces of information.

In the context of STAR-CITY, there are many ontology streams, each capturing a different aspect of traffic-related data at a given time instant. For example, one ontology stream captures weather information, and another captures real-time bus positions.

One interesting task of the system is pattern association mining. This module is responsible for generating predictions based on semantically-described historical data. New approaches to mine association rule had to be proposed since all data in the STAR-CITY system is represented as ontology streams.

In specific, an algorithm to mine Description Logics rules was proposed. The technical and theoretical aspects of learning such rules are discussed mainly in Lécué and Pan (2013), Lécué (2015) and Lécué and Pan (2015). The usage of such works as a module to predict the severity of road traffic congestion is described in Lécué et al. (2014b). Together with other modules, the whole STAR-CITY project is discussed in Lécué et al. (2014a) and Lécué et al. (2014c).

Here, we summarize DL rule mining's main ideas, as discussed in Lécué and Pan (2013). Each ontology snapshot can be viewed as a transaction. The ontology stream is, therefore, a time-ordered sequence of transactions. A DL association rule is an association between knowledge from two different ontology streams at a given time point.

Let O and P be two ontology streams, and O^n, P^n represent their snapshots at time n. A DL association rule between O and P associates knowledge from O^i to predict knowledge in P^i. This means that the rule's body contains only facts from O^i, and the rule's head contains only facts from P^i.

For example, consider that O is the ontology stream that contains all context knowledge, such as sensor readings and weather data, and consider that P contains knowledge about each road congestion's severity. DL association rules from O to P can be learned using historical data and then be used to (i) understand what causes road congestion, (ii) predict road congestion using the knowledge from O about the current timestamp.

In this context, the learned rules combine different data sources to predict bus congestion roads. One example of DL rule in the STAR-CITY project is Rule 2.1. It can be read as "the traffic flow of road r_1 is heavy if r_1 is adjacent to a road r_2 where an accident occurred and the humidity is optimum." As can be noted, DL rules do not model time relations but can connect knowledge from many different sources, such as journey times, social media, and weather information streams (LÉCUÉ et al., 2014a).

$$\begin{aligned} HeavyTrafficFlow(s) \leftarrow & Road(r_1), Road(r_2), \\ & isAdjacentTo(r_1, r_2), \\ & hasTravelTimeStatus(r_1, s), \\ & hasWeatherPhenomenon(r_1, w), \\ & OptimumHumidity(w), \\ & hasTrafficPhenomenon(r_2, a), \\ & RoadTrafficAccident(a) \end{aligned} \qquad \text{(Rule 2.1)}$$

DL rules mining has also been reportedly applied to a different domain. In Lécué and Wu (2017), the mined rules are used to explain and predict abnormal travel expenses on Accenture employees' data. In this case, rules were extracted to explain why a given expense can be considered abnormal, considering semantic contextual information such as where, when, and what was happening when the expense occurred.

Although the algorithms for mining DL rules are discussed in the papers, there is no implementation available. Moreover, the STAR-CITY data pipeline is built using private data management components.

2.4.4 Other approaches

In this subsection, we review some other approaches for mining RDF-graphs. They explore mining rules with interesting properties, such as taking advantage of class taxonomy and multiple head relations. In general, they are limited to mine very specific patterns.

Multi-relation Association Rules

Ramezani et al. (2014) propose another data mining task, called *Multi-relation Association Rules Mining*, and an algorithm, MRAR, capable of mining them. In their definition, a *Multi-relation Association Rule* means "those rules that have more than one relation in at least one of their items." This would be different from the definition they use of *Multi-relational Association Rules*, where the "rules [...] are extracted from multiple tables (multiple relations)."

An example of association rule mined by Ramezani et al. (2014) is the rule expressed in Rule 2.2. It indicates that "those who live in a place which is near to a city with humid climate type and also are younger than 20 also have a good health condition" (RAMEZANI et al., 2014). Just as in association rules, support and confidence metrics can be associated with the rule.

$$\begin{aligned} & LiveIn(NearTo(ClimateType(Humid))), \\ & \quad AgeLessThan(20) \Rightarrow HealthCondition(Good) \end{aligned} \qquad \text{(Rule 2.2)}$$

We argue that Rule 2.2 could be alternatively written on an equivalent first-order rule notation, as expressed in Rule 2.3. As we understand, traditional Multi-relational Association Rules, and AMIE, should potentially mine this rule.

$$\begin{gathered} LiveIn(x,y),\ NearTo(y,z),\ ClimateType(z,Humid), \\ AgeLessThan(x,20) \Rightarrow HealthCondition(x,Good) \end{gathered} \tag{Rule 2.3}$$

MRAR can also mine rules with multi-relations in the rule's head, such as in Rule 2.4. This rule can also be represented in a first-order logic notation, but it could not be mined by algorithms such as AMIE, which predicts a single relation atom.

$$HealthCondition(Good) \Rightarrow LiveIn(Near(ClimateType(Humid))) \tag{Rule 2.4}$$

$$\begin{gathered} HealthCondition(x,Good) \Rightarrow LiveIn(x,y), \\ Near(y,z),\ ClimateType(z,Humid) \end{gathered} \tag{Rule 2.5}$$

The approach proposed by Ramezani et al. (2014) introduces new concepts for rule mining, which are aggregated in the novel algorithm MRAR. It is a modification of Apriori that mines frequent itemsets by traversing the input RDF-graph recursively.

MRAR defines an important concept called *ItemChain*, analogous to the itemset concept in Apriori. An *ItemChain* represents a path from an initial entity to a target entity in the input graph. It can be represented by the list of relations required to go from one concept to another. An *ItemChain's* support can be defined as the number of entities such that, following the *ItemChain's* relations, gets to the *ItemChain's* target.

L-Large ItemChains are defined in a similar way to Apriori, by combining two *(L-1)-Large ItemChains*. Specific data structures are defined to keep track of all information required to efficiently generate *L-Large ItemChains* and evaluate their metrics.

The mined rules may have multiple *ItemChains* in the antecedent, and exactly one *ItemChain* as the consequent. The imposed limitation on the consequent is intended to limit the volume of generated rules (RAMEZANI et al., 2014). Each *ItemChain* may contain multiple relations.

Similarly to AMIE, a relation r is considered to be an r-attribute of its subject. MRAR mining strategy also imposes that all rule's *ItemChains* have the same starting node, which can be considered an inconvenient limitation in the search space. For example, the rule $r1(x,y),\ r2(y,A),\ r3(y,B) \Rightarrow r4(x,C)$ cannot be properly represented as an MRAR rule. For example, a tentative to represent it as $r1(r2(A)), r1(r3(B)) \Rightarrow r4(C)$ actually does not guarantee that relations $r2$ and $r3$ states facts about the same entity y.

de Oliveira et al. (2019) recently proposed an extension to MRAR, called MRAR+. MRAR+ focuses on taking advantage of interlinked knowledge, such as in the Linked Open Data, without sacrificing mining performance.

The algorithm is divided in a 4-step process. In Step 1, MRAR is applied to a single RDF-graph. It is expected that this RDF-graph, as part of the Linked Data, has entities connected to other datasets.

After mining the main RDF-graph, Step 2 selects a subset of the other KBs available. The sub-sets are selected by considering the entities mined in the first step that are linked to other KBs' data. Step 3 expands the original database by adding the selected knowledge and Step 4 reruns MRAR. At this moment, the representation being mined can better capture the context of the entities and relations, potentially generating more interesting rules.

MRAR+ is executed on some test data, and the comparison of the rules mined on Step 1 and the ones mined on Step 4 shows that the second set of rules indeed used the extended knowledge to represent more complex patterns. The same rules could be mined if all KBs were considered in Step 1 but at the cost of having unnecessary data and being much slower. MRAR+ is available to download on GitHub[14].

SWARM algorithm

Barati et al. (2016) investigate the limitations of AMIE and other KB mining algorithms regarding the usage of schema-level knowledge during mining. They argue that previous approaches explore only instance-level data, ignoring the semantics of schema-level. They propose the SWARM (Semantic Web Association Rule Mining) algorithm which is able to consider *rdf:type* and *rdf:subClassOf* relations.

SWARM mines rules in the form $r1(x, C) \Rightarrow r2(x, B)$. Moreover, they group similar rules based on the values that x may assume, such as shown in Rule 2.6. The rule states that *John Lennon* and *George Harrison* are examples of entities for the rule "if x has an *instrument guitar*, then x has the *occupation songwriter*." The rule can be alternatively represented as in Rule 2.7.

$$\{John\ Lennon,\ George\ Harrison\} : (instrument,\ Guitar) \Rightarrow (occupation,\ Songwriter) \tag{Rule 2.6}$$

$$instrument(x,\ Guitar) \Rightarrow occupation(x,\ Songwriter), x \in \{John\ Lennon,\ George\ Harrison\} \tag{Rule 2.7}$$

Furthermore, SWARM uses the *rdf:type* and *rdf:subClassOf* relations in the KB's schema to better assess the rule's support and confidence metrics. Using these relations and the class taxonomy they entail, the variable's class can be adjusted to maximize the rule's metrics. Rule

[14] <https://github.com/feliperj629/MRAR_plus>

2.8 shows an example of a rule using schema-level knowledge, stating that the person is a $Scientist$ yields better confidence than if it was unspecified or generalized as a $Person$.

$$\{Wallace,\ Darwin,\ Wiles,\ Bunsen\}\{Scientist\} : (knownFor,\ Natural\ selection) \\ \Rightarrow (award,\ Copley\ Medal) \quad \text{(Rule 2.8)}$$

Neural networks

Machine Learning is a sub-field of Artificial Intelligence that has received significant attention in the last decade because of the continuous state-of-the-art results in many research fields and applications. Data and algorithms are brought together to solve pattern recognition, learning, and decision-making problems (STOICA et al., 2017). Many of these recent advancements are based on deep learning methods.

One way to use deep learning models is to explore their representation-learning by mapping complex data to numerical vectors while also preserving many intrinsic data characteristics. *word2vec* is a well-known model that efficiently learns high-quality representations that capture a large number of syntactic and semantic relations between words (MIKOLOV et al., 2013). The model loops on the words of a set of sentences, using the current word to predict its neighbors. After trained, the neural model's internal representation for each word can be used as an embedding of the word, i.e., a mapping transformation from a token to a vector space.

Approaches for mining association rules using neural networks have been investigated (SUCHANEK et al., 2019). For example, Omran et al. (2018) learn predicate embeddings P_i and use these embeddings to find potential rules. The intuition is that, given the rule $P_1(x, z) \wedge P_2(z, w) \wedge \ldots \wedge P_n(v, y) \rightarrow P_t(x, y)$, the embedding of $P_1 \cdot P_2 \cdot \ldots \cdot P_n$ must be similar to P_t. They compare their method with AMIE+ and show that their approach outperforms AMIE both in time efficiency and the number of mined quality rules.

All discussed approaches for mining rules from KB data are interesting and tackle the problem from a different requirements perspective. Table 2.3 summarizes the different rule mining approaches discussed in this chapter. We note that, unfortunately, most of the algorithms are not publicly available.

Table 2.3 – Comparative table of different KB rule learning approaches

Approach	**Algorithm**	**Data input representation**	**Metrics**	**Implementation available**
Multi-relational association rule	WARMR (DEHASPE; RAEDT, 1997)	Relational data	Support, Confidence	Yes
Multi-relation association rule	MRAR (RAMEZANI et al., 2014) MRAR+ (de Oliveira et al., 2019)	RDF-Graph RDF-Graphs	Support, Confidence	No Yes
Semantic Association Rule	SWARM (BARATI et al., 2016)	RDF-Graph	Schema-aware Support and Confidence	No
Embedding-based rules	RLvLR (OMRAN et al., 2018)	RDF-Graph	Head Coverage, Standard Confidence	No[15]
Horn rules	AMIE 3 (LAJUS et al., 2020)	RDF-Graph	Head Coverage, PCA Confidence	Yes
DL rules	Lecue's (LÉCUÉ; PAN, 2013)	Ontology Stream	Support, Confidence	No

[15] The executables are available without documentation or source code at <https://www.ict.griffith.edu.au/aist/RLvLR/>.

Chapter 3
Mining rules from ontology-based trajectories

This chapter discusses our approach for mining rules from ontology-based trajectories using the KB rule mining algorithm AMIE. This research is based on the following observations:

1. As reviewed in Section 2.3, there are multiple ontology-based trajectory representations proposed in the literature. The recently proposed MASTER, although not explicitly modeled as an ontology, also uses RDF as its underlying logical representation. Therefore, all these representations can be processed as an RDF-graph.

2. Trajectory data contains facts about real-world entities and their interactions. Specifically, it represents how moving objects interact with physical places and how all these entities are connected with different semantic aspects.

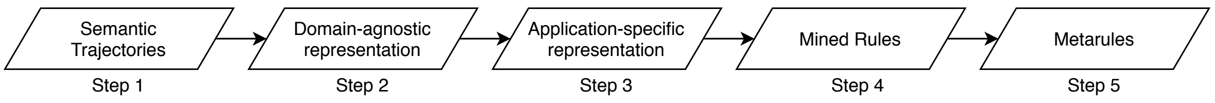
Therefore, we consider that ontology-based semantic trajectory representations are Knowledge Bases suitable to be mined by KB rule mining algorithms. Based on the literature review on ontology-based trajectory representations (Section 2.3) and KB rule mining algorithms (Section 2.4), we limit this investigation's scope to use:

- *The STEP ontology as our data representation.* STEP's main advantage is its online availability and documentation.

- *AMIE 3 as our logical rule mining algorithm.* AMIE 3 is a well-known, publicly available framework for mining rules from RDF-graph. It represents the state-of-the-art, considering its performance compared to other algorithms, and allows implementing different mining bias.

Section 3.1 discusses the proposed data pipeline at a high level. Section 3.2 through Section 3.6 discuss in-depth details of each pipeline step. We include in each section a Minimum Working Example, which is based on a hand-crafted LBSN dataset. Section 3.7 discuss how

Figure 3.1 – Our proposed data pipeline for logical rule mining.



mined knowledge may be used. Finally, Section 3.8 discusses how our rules distinguish from current rules mined in the trajectory data mining community.

3.1 Pipeline overview

In this work, we propose a 5-step data pipeline, as illustrated in Figure 3.1. This pipeline is similar to the data mining pipeline proposed in Fayyad et al. (1996), and to trajectory mining frameworks such as the ones proposed in Albanna et al. (2015) and Feng and Zhu (2016). Semantic trajectory data must be acquired, represented, transformed, mined, pos-processed, and finally used in different applications.

The first step in this pipeline is composed of acquiring, building, and selecting semantically-enriched trajectory data. Multiple algorithms and frameworks can be used in this step, depending on the application requirements and data availability. These approaches have been extensively investigated in the literature and were discussed in Section 2.3.

Next, semantic trajectories are represented and stored in an ontology-based representation, such as *Baquara*[2], STEP, or MASTER. In this work, we focus on STEP representation, which is readily available on the internet and is well-documented. More specifically, we use a subset of STEP since we do not represent raw trajectories or spatio-temporal data.

Ontologies proposed in the literature are built to support multiple trajectory data in the context of movement analysis. Each of them offers different ways to extend data to meet different requirements. Nonetheless, extending the relations in the T-box is not a common practice.

Since we aim to mine relational rules, we argue in favor of a dynamic ontology-based representation built from a populated STEP ontology. We propose in Step 3 a systematical approach to convert STEP to an application-specific representation, which explicitly represents relationships as relations.

Finally, Step 4 applies AMIE to mine rules. We detect and discuss multiple limitations and characteristics that make the off-the-shelf algorithm unsuitable for mining rules from trajectory data. We propose some modifications that allow us to mine more interesting rules than those previously mined in the literature.

To improve data analysis, Step 5 of our pipeline uses an approach to group similar rules, based on the concept of metarules. This post-processing step greatly reduces the number of mined rules by finding general pattern templates.

3.2 Semantic Trajectories

Trajectory conceptualizations and representations have different terminologies and allow different semantic data abstractions. Some general concepts can be identified, since there is basically at least one *moving object* executing a *trajectory* by sequentially *visiting* a set of *places*. Each of these can be associated with multiple semantics and between themselves.

As discussed in Section 2.3, GPS-based data can be processed and transformed in Semantic Trajectory, with concepts such as point of interest (POI), and stops and moves. Nonetheless, this requires the application of many algorithms, from raw data filtering to episode extraction and annotation.

Instead of relying on such algorithms, we focus on this work in using LBSN datasets, which include users' check-ins in real-world places. This kind of data is inherently annotated with semantics, like venue names and categories. These semantics can also be used to integrate knowledge from external data sources. LBSN datasets contain no move information, and we cannot know exactly when a user arrived and departed from a given place.

We consider from now on the main concepts of `User`, `Trajectory`, `Check-in`, and `Venue`. Nonetheless, the proposed approach can be applied to any semantic trajectory data represented in STEP.

All described concepts can be associated with qualitative semantic descriptions, as well as with each other. For example, a venue can be associated with a venue category, a *hasFriend* relationship can associate user entities, and a *withinTimeWindow* relation can connect two check-ins that occur within a time window defined by a user-specified time threshold.

Any qualitative semantic data can be used, such as weather or check-in-related tags. As discussed in subsection 2.4.2, the limitations imposed by the proposed rule-mining strategy are twofold: numerical data should be transformed into categorical data, and hierarchies cannot be mined. Based on this, we do not represent raw spatiotemporal data or venues' categories taxonomy.

Finally, it is essential to note that the flexibility in representing inter-concept relations must be analyzed on a case-by-case basis. Which patterns can be mined is directly dependent on how and which data is represented.

3.2.1 Data incompleteness

Data incompleteness can be motivated by multiple factors, depending on the data sources used and the data processing applied to them. In the case of LBSN data, not all users' visits to real places are available as digital footprints. Inferring unknown places is out of the scope of this research. Nonetheless, other data incompleteness assumptions can be made. For example, we might not know a recently-opened venue's category.

As discussed in Section 2.4, there is much research in (in)completeness-aware measures in the context of Knowledge Bases. For example, AMIE uses the Partial-Completeness Assumption and the associated PCA Confidence.

Here, we propose a set of possible assumptions that may apply to the LBSN datasets investigated. We simulate these incompleteness processes in our datasets by sampling data during the mining phase. This allows an initial discussion about PCA and PCA Confidence in the context of trajectory data but should be further extended in future works.

Consider the following data incompleteness sources:

- Venue-related: Suppose that the check-in's venue is obtained by checking the check-in's geo-location against a Venues dataset. Therefore, the check-in's geo-location resolution might not be enough to determine the exact venue associated with it, or we might not have any venue registered at that location in our dataset. Complementary, we might not know all semantics related to a venue, such as its venue category.
- Time-related: regarding the timestamp associated with a check-in, we consider that we might also not have full-time granularity. For example, the specific hour of a check-in might be unknown.
- Friendship: friendship links form a dynamic network with frequent updates. Therefore, we probably cannot suppose that we have all the real friends of a user at a given time.

Note that PCA may or may not hold in these different cases. For example, if a venue can have at most one category, then PCA holds since we will have a functional relation. On the other hand, if a venue can have multiple categories, then the PCA degrades as the set of categories combinations increases. Finally, PCA does not hold for friendship links.

3.2.2 Minimum Working Example

Throughout this chapter, we use a small yet illustrative dataset to discuss this work's strengths and weaknesses. The data was manually crafted and is described in Table 3.1. It shows the trajectory of three people, Bob, Mary, and Jen.

Bob has two trajectories, one on a Thursday and another on a Friday. All of his check-ins occurred during the morning. As an example of concepts relations, we consider in this research that the time window in which check-ins occurred might be an interesting event defined by a domain expert. Consider, for example, the check-ins 1_1_1 and 1_1_2, which occurred within a 2-hour time window. Therefore, we connect these two check-ins with a "within time window" relation ($withinTimeWindow$).

Mary and Jen also have their own trajectories and check-ins. Note, for example, that all three users checked-in at the restaurant $Restaurant\_0$.

Table 3.1 – Example of an LBSN dataset. We have a unique identifier for Users, Trajectories, Check-ins, and Venues, as well as semantic data about the venue's categories and check-in date and time. We might also connect check-ins that occur within a time window of, in this example, 2-hour length.

User ID	**Trajectory ID**	**Check-in ID**	**Venue id**	**Venue category**	**Date/time**	**Within Time Window**
Bob	1_1	1_1_1	Bob's Home	Home	Thu 9:00	{1_1_2}
		1_1_2	Restaurant_0	Restaurant	Thu 10:00	{1_1_1}
	1_2	1_2_1	Office_0	Office	Fri 10:00	{}
Mary	2_1	2_1_1	Restaurant_0	Restaurant	Thu 14:00	{}
	2_2	2_2_1	Office_0	Office	Fri 15:00	{}
Jen	3_1	3_1_1	Restaurant_0	Restaurant	Sat 13:30	{}

Table 3.2 – Example of a LBSN friendship dataset. We consider that friendship is a symmetric relation. This means that if user u is friend of user v, v is also friend of u.

u	**v**
Bob	Mary
Mary	Jen

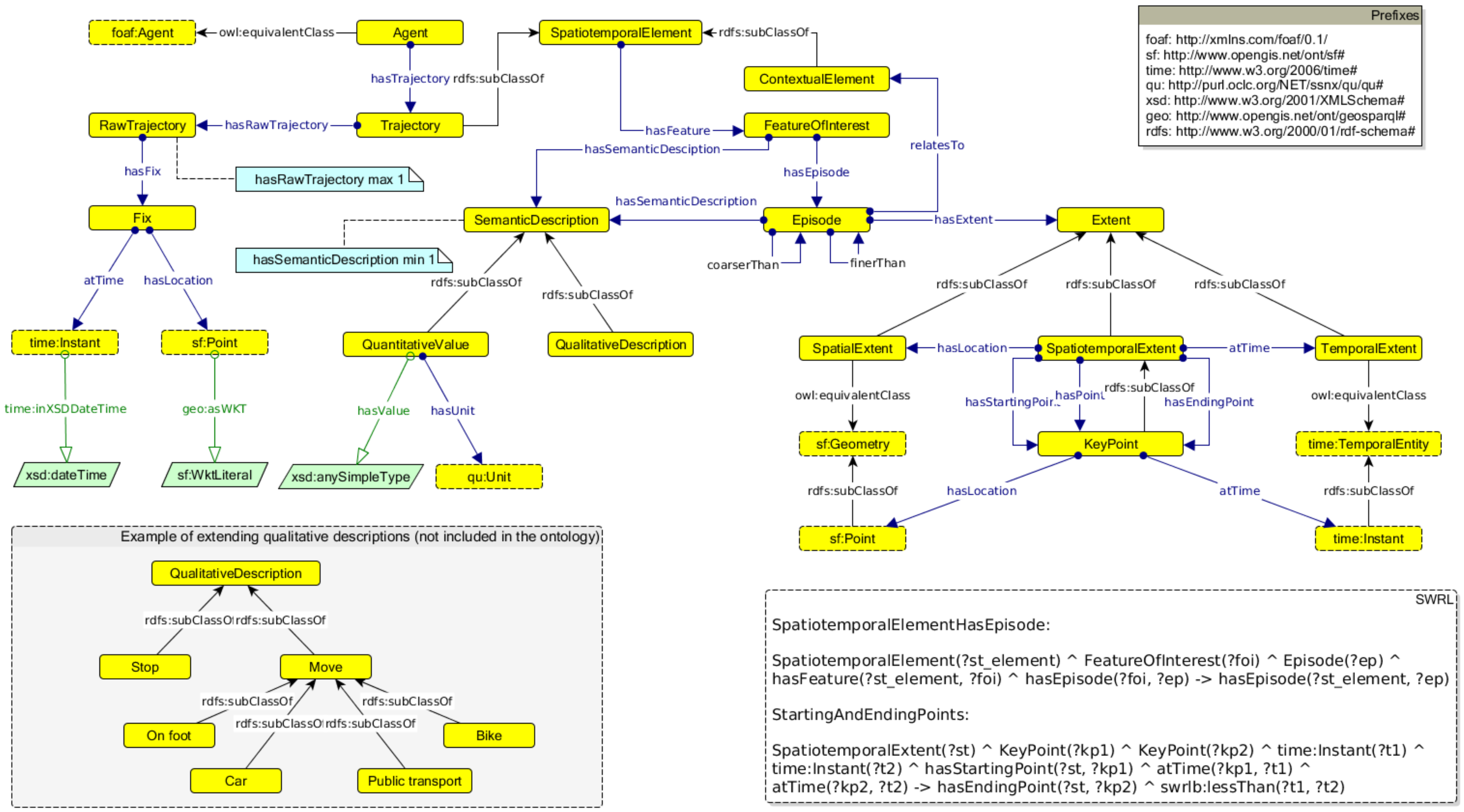
Finally, Table 3.2 shows us the relationship between these users. Mary is a friend of Bob and Jen, but Bob and Jen are not each other's friends. We consider that friendship is a symmetric relation, i.e., if Mary is a friend of Bob, Bob is also a friend of Jen.

3.3 Ontology-based representation

Knowledge Engineering and Ontology Design Patterns are both broad research fields concerned with the best practices of building knowledge representations and ontologies (GANGEMI, 2005; HU et al., 2013). Semantic trajectory ontologies such as *Baquara*[2] and *STEP* loosely follow such practices, yielding data representations able to capture recurring issues in cross-domain projects (HU et al., 2013; FILETO et al., 2015b; NOGUEIRA et al., 2018).

As discussed at the beginning of this chapter, we focus on STEP representation, mainly because of its online availability. Figure 3.2 depicts the STEP ontology schema as proposed in Nogueira et al. (2018), including possible concept extensions and inference rules.

Figure 3.2 – STEP ontology. Source: Nogueira et al. (2018).



```
foaf: http://xmlns.com/foaf/0.1/
sf: http://www.opengis.net/ont/sf#
time: http://www.w3.org/2006/time#
qu: http://purl.oclc.org/NET/ssnx/qu/qu#
xsd: http://www.w3.org/2001/XMLSchema#
geo: http://www.opengis.net/ont/geosparql#
rdfs: http://www.w3.org/2000/01/rdf-schema#
```

```
SpatiotemporalElementHasEpisode:

SpatiotemporalElement(?st_element) ^ FeatureOfInterest(?foi) ^ Episode(?ep) ^
hasFeature(?st_element, ?foi) ^ hasEpisode(?foi, ?ep) -> hasEpisode(?st_element, ?ep)

StartingAndEndingPoints:

SpatiotemporalExtent(?st) ^ KeyPoint(?kp1) ^ KeyPoint(?kp2) ^ time:Instant(?t1) ^
time:Instant(?t2) ^ hasStartingPoint(?st, ?kp1) ^ atTime(?kp1, ?t1) ^
atTime(?kp2, ?t2) -> hasEndingPoint(?st, ?kp2) ^ swrlb:lessThan(?t1, ?t2)
```

In STEP, a moving object (`Agent`) is associated with multiple trajectories, and each `Trajectory` can be associated with `Semantic Description’s` through a `Feature Of Interest` (`FOI`) instance. `Semantic Description’s` can be quantitative or qualitative, and are encouraged to be extended to specific use cases.

A trajectory may also be connected to `Episode’s` through an associated `FOI`. An `Episode` in STEP is the smallest semantic unity available. It encapsulates values that a `Feature of Interest` or `Contextual Element` may assume during the trajectory (NOGUEIRA et al., 2018).

Raw trajectory data and raw spatiotemporal data can be represented using `RawTrajectory` and `Extent` entities. Also, an `Episode` can be connected to `ContextualElement’s`, which allows a generic way to group multiple contextual features.

Nogueira et al. (2018) discuss how the STEP ontology can be extended. Two main examples are provided, and are also present in Figure 3.2. The first one is regarding the T-Box, where new classes can be created. The example given shows how one might extend the `QualitativeDescription` concept to accommodate the `Stop` and `Move` concepts, and then refine `Move` with different movement strategies such as `On foot`, `Car` or `Public transport`.

The second possible extension to STEP is through SWRL[1] rules. One of the rules embedded in STEP is the rule *SpatiotemporalElementHasEpisode*, represented as Rule 3.1. This rule directly connects a `SpatiotemporalElement` to a previously indirectly connected `Episode`, bypassing the intermediate `FeatureOfInterest` entity.

$$hasFeature(st\_element, foi),\ hasEpisode(foi, ep) \Rightarrow hasEpisode(st\_element, ep) \quad \text{(Rule 3.1)}$$

The existence of such a rule shows that, for some applications, it might be more convenient to compress the ontology, allowing fewer hops between important concepts. We take this idea one step further in the next section by proposing a dynamically-built compressed version of STEP.

Complementary, it is important to note that how data is represented is intrinsically related to which patterns can be mined. Therefore, each decision on data representation affects whether we can mine certain patterns and how easily such patterns can be mined. Different applications might benefit from different representations.

In the context of this work, we consider that a *step:Agent* (`User`) executes a *step:Trajectory* (`Trajectory`) by visiting a set of *step:Episode*’s (`Check-in’s`) associated to a *step:ContextualElement* (`Venue`). The semantic data associated to the different concepts are represented as a combination of `FeatureOfInterest’s` and `QualitativeDescription’s`.

[1] Semantic Web Rule Language, <https://www.w3.org/Submission/SWRL/>.

We extend the STEP ontology by adding subclasses of `FOI` for each of our usages. For example, to represent a venue's category, we create a `VenueCategory FOI` class.

Finally, to connect two episodes, which in our case represent check-ins, we extend the `ContextualElement` with a `ContextualRelations` subclass. We proceed to create one FOI for each relationship. For example, if we want to connect check-ins **A** and **B** using a *withinTimeWindow* relationship, we create a `WithinTimeWindow FOI` instance, which connects **A**'s `ContextualRelations` instance and **B**. A concrete example is given below, in subsection 3.3.1.

Note that although the `Episode` concept has a broad meaning in STEP, we use it solely to represent `Check-in's`. The mixture of elements such as `Feature of Interest's`, `Semantic Description's` and `Contextual Element's` makes it hard to have a single and clear approach to represent data. At the best of our abilities, we have represented the available data in a way that respects the STEP ontology while also making it easier for data processing in our pipeline.

Raw trajectory data and the exact time and location of the trajectory are not of interest in this dissertation's scope. The same applies to quantitative data since we cannot properly represent numeric values in our rules. Therefore, we do not use the related concepts of `RawTrajectory`, `Extent` and `QuantitativeValue` defined in STEP.

3.3.1 Minimum Working Example

We represent the *Trajectory 1_1* from Table 3.1 in Figure 3.3.

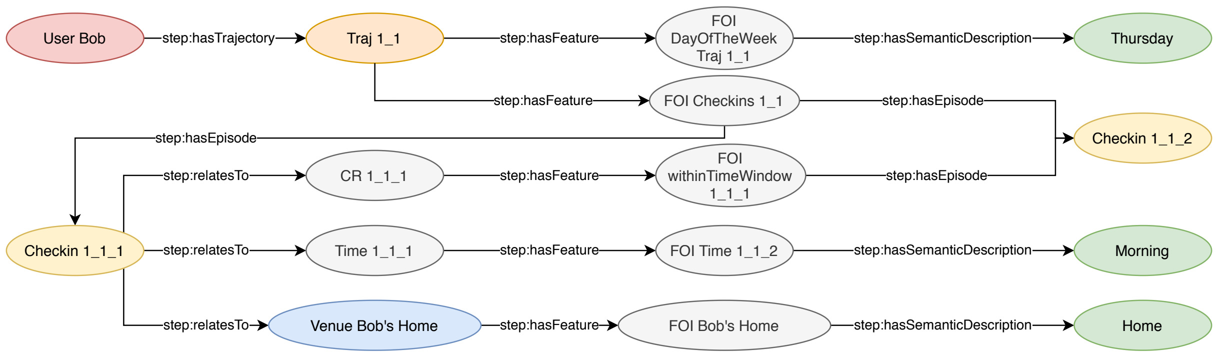


Figure 3.3 – Partial representation of STEP (Figure 3.2) populated with the data from Table 3.1. Only trajectory 1_1's data is fully shown. Same colors represent similar concepts for visual guidance. The gray nodes represent uninteresting entities which are discussed in Section 3.4.

Figure 3.4 shows the extended subset of the STEP's T-box. This extension includes a set of classes dynamically added while populating the ontology based on the available data samantics.

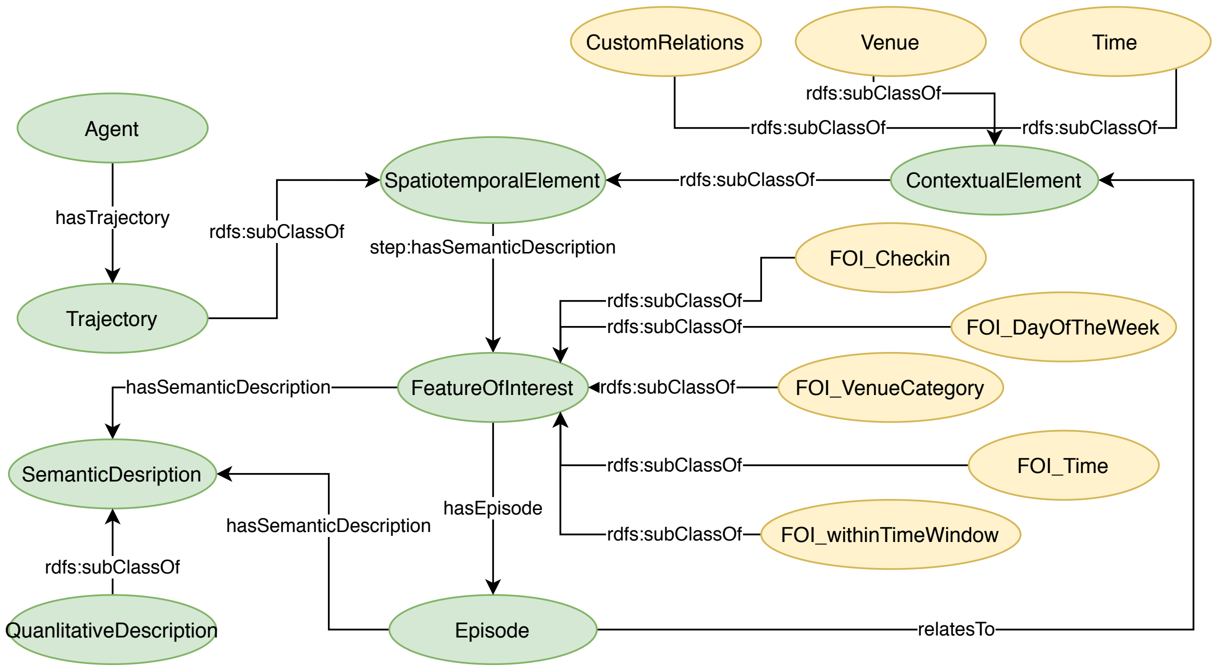


Figure 3.4 – STEP's T-Box extended when creating the ontology from Figure 3.3. Green nodes represent STEP concepts, while yellow nodes show classes added when processing semantic data. STEP's entities that are not used in this work are omitted.

3.4 Application-specific representation

Consider again the populated STEP ontology represented in Figure 3.3. Note how the STEP representation requires many *intermediate* nodes, composed mainly by `FOI` entities. Querying this data may be made easier by using inference rules such as *SpatiotemporalElementHasEpisode* (Rule 3.1).

Even so, we argue that STEP and other domain-independent representations proposed in the literature are not well suited for being mined for logical rules. Indeed, mining Horn rules from these representations is very hard since most semantic relationships are encoded as instances and classes, not relations. For example, STEP heavily depends on extending classes and creating `FOI` instances.

We consider that these semantic trajectory ontologies must be transformed into a simplified and application-specific representation.

Although the process of defining such simplified representation has to be manually and carefully done, the transformation from ontologies like *STEP* to a simplified ontology should be easily automated. More specifically, we take advantage of the class extensions introduced in the previous section to dynamically and systematically build an application-specific data representation.

3.4.1 Mapping from STEP to Application-specific ontology

The main goal of changing the ontology-based representation is to transform semantic relationships encoded as `step:FOI's` instances into explicit relations. From now on, we refer to STEP's concepts using the prefix `step:` and to the concepts in our dynamically built representation with `converted:`.

We start by mapping concepts that has a one to one mapping. They include transforming `step:Agent` into `converted:User`, `step:Trajectory` into `converted:Trajectory`, and `Episode` into `converted:Check-in`. Next, we transform `FOI's` subclasses into relations. For example, the `FOI` subclass `step:VenueCategory FOI` becomes the relation *converted:hasVenueCategory*, and the entities and literals connected by the `FOI` instances are now connected by the associated relation.

Since a `step:ContextualElement` instance may represent a `Venue` or a `ContextualRelations`, we implement two different behaviors for its convertion. If the `step:ContextualElement's` represents a *Venue*, we create the venue entity `converted:Venue` and apply the `FOI` transformation previously described. Otherwise, we ignore the contextual element, and connect the check-in directly to the `FOI's` connected to the `step:ContextualElement`, and then apply the previous process.

Next, we show the pseudocode for our conversion from a populated STEP to our dynamic-generated representation. The code is split into multiple parts, where Algorithm 3.1 specifies the main one, which accepts as input a STEP ontology with all facts of interest. Its base idea is to process the ontology-graph in an orderly fashion, similar to a depth-first search. It begins by processing an `step:Agent`, which triggers the processing of the agent's trajectories, and then the trajectory's `FOI's`. Processing an `FOI` means processing its associated `SemanticDescription's` and `Episode's`. At each step, new entities, relations, and relationships are added to the converted representation.

Algorithm 3.1 uses Algorithm 3.2 to transform a `step:SemanticDescription` entity into a relation. It uses the intermediary `FOI's` class (the `FOI` which is between the `step:SpatiotemporaElement` and the `step:SemanticDescription`) as the new relation name, and the `step:SemanticDescription`'s value as the relation's object.

When processing `step:Episode's`, Algorithm 3.1 uses Algorithm 3.3 to process the episode's contextual elements. It implements different logics depending whether the contextual element is a `Venue` or not. If it represents a `Venue`, we create a `converted:Venue` instance and connect the check-in to it with *converted:hasVenue*. All venue's `FOI's` are processed in order to enrich the venue with its associated `SemanticDescription's` values.

If the `ContextualElement` is not a `Venue`, then it is not mapped as an entity. Instead, its associated `FOI's` are treated as directly connected to the original `Episode`. If such `FOI` has an associated `Episode`, this spawns a direct link between two `Episode's`. If

Algorithm 3.1 STEP to application-specific representation. Given a STEP ontology, dynamically builds a converted representation.

```
1: procedure STEP2CONVERTED(STEP-based trajectory data: $STEP$)
2:     for each $agent$ in $STEP$ do
3:         $converted\_agent \leftarrow$ Map $agent$ from $step{:}Agent$ to $converted{:}User$
4:         for each $traj$ in $agent.hasTrajectory$ do
5:             $converted\_traj \leftarrow$ Map $traj$ from $step{:}Trajectory$ to $converted{:}Trajectory$
6:             Add fact $converted{:}hasTrajectory(converted\_agent,\ converted\_traj)$
7:             for each $foi$ in $traj.hasFeature$ do
8:                 $FOI\_name \leftarrow foi's\ class\ name$
9:                 for each $sem\_desc$ in $foi.hasSemanticDescription$ do
10:                    SEMDESC2RELATION($custom\_traj$, $FOI\_name$, $sem\_desc$)
11:                for each $episode$ in $foi.hasEpisode$ do
12:                    $checkin \leftarrow$ Map $episode$ from $step{:}Episode$ to $converted{:}Checkin$
13:                    Add fact $converted{:}has\{\$FOI\_name\}(custom\_traj,\ checkin)$
14:                    PROCESSCONTEXTUALELEMENTS($converted\_checkin$,
                                                  $checkin.relatesTo$)
```

Algorithm 3.2 Convert `step:SemanticDescription` to relation. Given an entity, a FOI name, and a qualitative description, the procedure adds a fact stating that the entity (subject) has a relation (FOI name) with the value of the qualitative description (object).

```
1: procedure SEMDESC2RELATION(Entity: $entity$, String: $FOI\_name$,
   step:QualitativeDescription: $sem\_desc$)
2:     Map $sem\_desc$ from $step{:}QualitativeDescription$ to $converted{:}\{\$FOI\_name\}$
3:     Add converted $converted{:}has\{\$FOI\_name\}(entity,\ sem\_desc)$
```

such `FOI` has an associated `SemanticDescription`, this spawns a semantic feature of the `Episode`.

This set of procedures are enough to build a dynamically-generated compact alternative to STEP, where each relation contains semantic information to be mined by the approach proposed in this work.

3.4.2 Minimum Working Example

Following our data pipeline, the STEP representation of Table 3.1's data can be transformed to our application-specific representation by applying Algorithm 3.1. Figure 3.5 shows a subset of the generated ontology, similar to Figure 3.3. In essence, the difference between the populated STEP and our converted ontology is that the gray nodes in STEP (Figure 3.3) have been collapsed in ours application-specific representation (Figure 3.5) as either an explicit or an implicit relation.

Complementary, we show in Figure 3.6 the full application-specific ontology, with all the data presented in Table 3.1. We have omitted the STEP's version since its size makes it hard to represent graphically.

Algorithm 3.3 Convert `step:Episode`'s `ContextualElement`'s. Given an episode, the procedure process its related `ContextualElement`'s. This includes mapping the `ContextualElement`'s semantic descriptions and its associated episodes.

```
1: procedure ProcessContextualElements($converted : Checkin$: $converted\_episode$, List[$step : ContextualElement$]: $CEs$)
2:     for each $ce$ in $CEs$ do
3:         if $ce$ is-a ‘$Venue$’ then
4:             $venue$ ← Map $ce$ from $step$:$ContextualElement$ to $converted$:$Venue$
5:             Add fact $converted$:$hasVenue(converted\_episode,\ venue)$
6:             for each $foi$ in $ce.hasFeature$ do
7:                 $FOI\_name$ ← $foi's\ class\ name$
8:                 for each $sem\_desc$ in $foi.hasSemanticDescription$ do
9:                     SemDesc2Relation($venue$, $FOI\_name$, $sem\_desc$)
10:        else
11:            for each $foi$ in $ce.hasFeature$ do
12:                $FOI\_name$ ← $foi's\ class\ name$
13:                for each $episode2$ in $foi.hasEpisode$ do
14:                    $converted\_episode2$ ← Map $episode2$ from $step$:$Episode$
                                                            to $converted$:$Checkin$
15:                    Add fact $converted$:$\{\$FOI\_name\}(converted\_episode,$
                                                            $converted\_episode2)$
16:                for each $sem\_desc$ in $foi.hasSemanticDescription$ do
17:                    SemDesc2Relation($custom\_episode$, $FOI\_name$, $sem\_desc$)
```

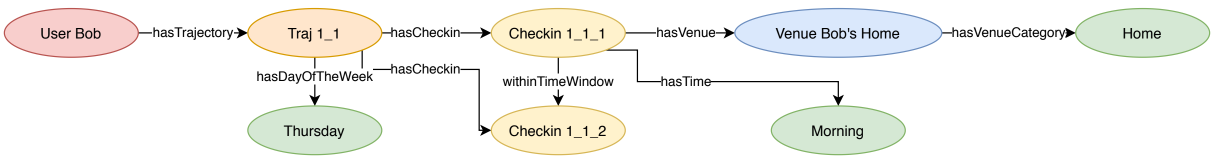


Figure 3.5 – Application-specific ontology generated by applying Algorithm 3.1 to the ontology from Figure 3.3. The same colors represent similar concepts for visual guidance.

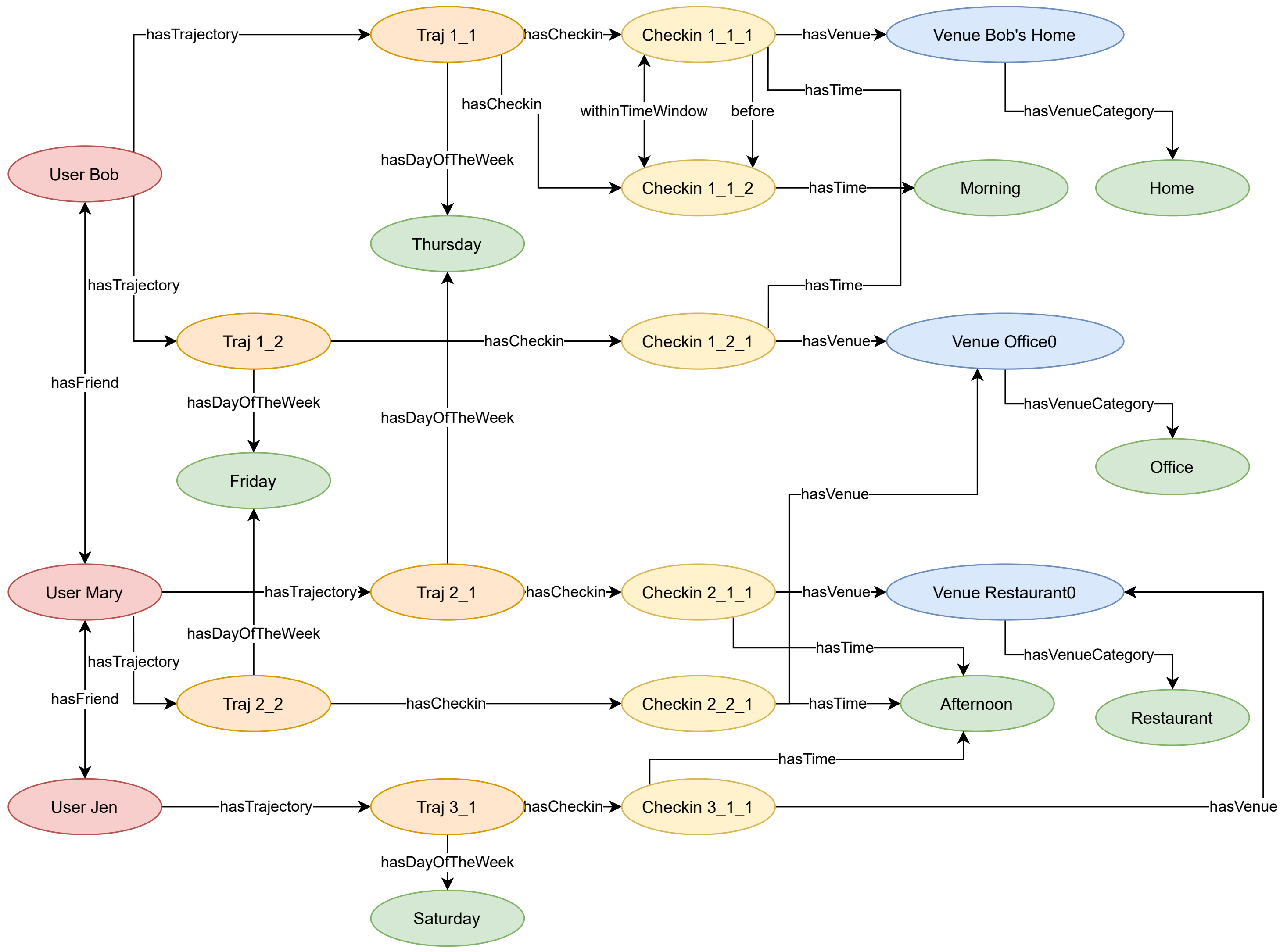


Figure 3.6 – Full application-specific ontology generated by representing the data from Table 3.1 in STEP and then applying Algorithm 3.1. The same colors represent similar concepts for visual guidance.

3.5 Rule mining

Once semantic trajectory data has been assembled and represented in an application-specific representation suitable for rule mining, we can finally apply AMIE to retrieve mobility patterns.

We start by applying the off-the-shelf AMIE 3 algorithm to the MWE dataset. We use these results to elaborate our subsequent discussion about different aspects of using a general-purpose KB rule mining algorithm to mine semantic trajectory data. The empirical discussion on mining the STEP representation is shown in Chapter 4.

More specifically, we choose to tackle three main issues found. By doing so, we also improve the AMIE 3 algorithm and its implementation. The mining issues we explore in this research are:

- **Uninteresting instantiated entities** (subsection 3.5.2): we observe that rules which describe specific trajectories or check-ins are uninteresting and can be therefore pruned during mining.
- **Injective mappings** (subsection 3.5.3): default AMIE does not enforce that different variables must be mapped to different constants. We use already implemented code to impose injective mappings and mine more interesting rules.
- **Symmetric-invariant rules** (subsection 3.5.4): We note that symmetric relations, e.g., *withinTimeWindow*, yields duplicated rules. We propose and implement pruning and filtering strategies which removes the duplicated rules.

3.5.1 Minimum Working Example

AMIE's data input is a set of RDF triples. Therefore, we export the converted ontology data in Figure 3.6 as triples, shown in Figure 3.7. We remove triples defining the ontology schema since we do not explore mining them in AMIE. The algorithm is then applied to mine rules with at most 4 atoms, with default parameters except by minimum initial support, set to 1, and recursion limit, set to 2.

AMIE mines 354 rules[3]. From these rules, 11 are 2-atom, 44 are 3-atom, and 299 are 4-atom. Next, we discuss some selected rules.

Rule 3.2 shows that AMIE mined a rule about the symmetric-property of the *hasFriend* relationship. This pattern may represent an interesting rule in many KBs, since the schema

[2] <https://www.w3.org/TR/n-triples/>

[3] The mined rules are available in our public code repository. See Chapter 4.

```
hasFriend(Bob, Mary)
hasFriend(Mary, Bob)

hasFriend(Mary, Jen)
hasFriend(Jen, Mary)

hasTrajectory(Bob, Traj_1_1)
hasTrajectory(Bob, Traj_1_2)

hasTrajectory(Mary, Traj_2_1)
hasTrajectory(Mary, Traj_2_2)

hasTrajectory(Jen, Traj_3_1)

hasDayOfTheWeek(Traj_1_1, Thursday)
hasCheckin(Traj_1_1, Checkin_1_1_1)
hasCheckin(Traj_1_1, Checkin_1_1_2)

hasDayOfTheWeek(Traj_1_2, Friday)
hasCheckin(Traj_1_2, Checkin_1_2_1)

hasDayOfTheWeek(Traj_2_1, Thursday)
hasCheckin(Traj_2_1, Checkin_2_1_1)

hasDayOfTheWeek(Traj_2_2, Friday)
hasCheckin(Traj_2_2, Checkin_2_2_1)
hasDayOfTheWeek(Traj_3_1, Saturday)
hasCheckin(Traj_3_1, Checkin_3_1_1)

hasTime(Checkin_1_1_1, Morning)
hasVenue(Checkin_1_1_1, Venue_Bob's Home)
hasVenueCategory(Venue_Bob's Home, Home)

hasTime(Checkin_1_1_2, Morning)
hasVenue(Checkin_1_1_2, Venue_Restaurant0)
hasVenueCategory(Venue_Restaurant0, Restaurant)

hasTime(Checkin_1_2_1, Morning)
hasVenue(Checkin_1_2_1, Venue_Office0)
hasVenueCategory(Venue_Office0, Office)

hasTime(Checkin_2_1_1, Afternoon)
hasVenue(Checkin_2_1_1, Venue_Restaurant0)

hasTime(Checkin_2_2_1, Afternoon)
hasVenue(Checkin_2_2_1, Venue_Office0)

hasTime(Checkin_3_1_1, Afternoon)
hasVenue(Checkin_3_1_1, Venue_Restaurant0)

withinTimeWindow(Checkin_1_1_1, Checkin_1_1_2)
withinTimeWindow(Checkin_1_1_2, Checkin_1_1_1)
```

Figure 3.7 – Triples generated from the ontology in Figure 3.6 are used as input to AMIE. Empty lines, triples sorting and facts' format were added for improved visualization. The actual format read by AMIE is similar to N-triples[2].

properties may not be known. In our case though, we are completely aware about the symmetric-property for our defined relations.

$$hasFriend(b, a) \Rightarrow hasFriend(a, b) \quad (HC\colon 1, Std.\ Conf.\colon 1) \tag{Rule 3.2}$$

The symmetric-relation unawareness may have even bigger issues. Consider for example Rule 3.3 and Rule 3.4. Although not yet obvious, these two rules state the same fact if we consider the symmetric-property of *hasFriend*[4]. We explore such duplicated rules in subsection 3.5.4.

$$hasFriend(b, c),\ hasTrajectory(a, d),\ hasTrajectory(c, d) \Rightarrow hasFriend(a, b) \quad (HC\colon 1, Std.\ Conf.\colon 1) \tag{Rule 3.3}$$

$$hasFriend(a, c),\ hasTrajectory(b, d),\ hasTrajectory(c, d) \Rightarrow hasFriend(a, b) \quad (HC\colon 1, Std.\ Conf.\colon 1) \tag{Rule 3.4}$$

Rule 3.5 shows a rule with an instantiated trajectory. We argue that rules with instantiated trajectories or check-ins are uninteresting, since they describe only a specific event. Moreover, a

[4] To show equivalence, one may apply the substitution $\sigma = \{a \mapsto b, b \mapsto a\}$ to Rule 3.3, and then change the consequent from $hasFriend(b,\ a)$ to $hasFriend(a,\ b)$. The result will be Rule 3.4.

trajectory and check-in are unique identifiers artificially generated to group similar concepts in an abstract view.

$$\begin{aligned} hasCheckin(Traj\_1\_1, c) \Rightarrow hasTime(c, Morning) \\ (HC\colon 0.33, Std.\ Conf.\colon 1) \end{aligned} \tag{Rule 3.5}$$

Rule 3.6 shows what we consider an interesting rule, since it connects two dimensions from check-ins: their time and their venue. This shows one of the flexibilities of mining Horn rules, since they can independently express the concepts.

$$\begin{aligned} hasTime(c, Afternoon) \Rightarrow hasVenue(c, Venue\_Restaurant0) \\ (HC\colon 0.33, Std.\ Conf.\colon 0.66) \end{aligned} \tag{Rule 3.6}$$

Of course, this flexibility also implies in a large set of possible combinations, yielding rules that mash up multiple aspects without actually providing interpretable and actionable patterns, such as Rule 3.7.

$$\begin{aligned} &hasFriend(u1, Mary),\ hasFriend(Mary, u2), \\ &\quad hasTrajectory(u2, t) \Rightarrow hasTrajectory(u1, t) \\ &\quad (HC\colon 0.60, Std.\ Conf.\colon 0.50) \end{aligned} \tag{Rule 3.7}$$

In the special case of Rule 3.7, the main issue is AMIE's default behavior of non-injective mappings. This means that, by default, AMIE does not enforce that different variables should be mapped to different instances. For example, Rule 3.7 predicts that both $u1$ and $u2$ have the same trajectory t, which we know by domain knowledge that must imply $u1$ to be equal to $u2$. Consequently, we have a rule which is uninteresting and hard to understand.

The same non-injective mapping problem occurs in Rule 3.8, but more indirectly. Note that check-ins $c1$ and $c2$ happen at the same venue and share a within time window check-in, $c3$. This is an indirect way to state that $c1$ and $c2$ are the same check-ins. Of course, the rule may be true for cases in which $c1 \neq c2$, but counting the cases in which they are equal is unfair since they contribute to artificially increasing the rule's confidence.

$$\begin{aligned} hasVenue(c1, p),\ hasVenue(c2{,}p),\ withinTimeWindow(c3, c2) \\ \Rightarrow withinTimeWindow(c1, c3) \\ (HC\colon 1, Std.\ Conf.\colon 0.50) \end{aligned} \tag{Rule 3.8}$$

Finally, Rule 3.9 is another example of an interesting rule we mine. It connects a trajectory's aspect (its day of the week) with a check-in's aspect (an specific venue).

$$\begin{aligned} hasCheckin(t, c),\ hasDayOfTheWeek(t, Friday) \Rightarrow hasVenue(c, Venue\_Office0) \\ (HC\colon 0.33, Std.\ Conf.\colon 1) \end{aligned} \tag{Rule 3.9}$$

As previously discussed, AMIE mined 354 rules in this example experiment, many of which are clearly uninteresting. Next, we discuss some of our approaches to reduce the number of such rules. We take the opportunity to anticipate their results by showing the number of rules mined after applying these techniques: subsection 3.5.2 reduces the number of rules from 354 to 286. When combined with subsection 3.5.3, we get 121 rules. Further adding subsection 3.5.4 reduces it to 80 mined rules.

3.5.2 Uninteresting instantiated entity

Consider the Rule 3.5. It describes that check-ins belonging to the Trajectory 1_1 in our data always occur during Morning. Consider also the handcrafted Rule 3.10.

$$hasVenue(CheckinX, p) \Rightarrow hasVenueCategory(p, Shopping) \tag{Rule 3.10}$$

Note that these rules describe specific facts regarding one specific trajectory or check-in. We consider such rules to be uninteresting since their knowledge cannot be generalized to other entities.

More specifically, we consider that a rule which states a fact about a specific trajectory or check-in is not interesting. In this context, we see the `Trajectory` and `Check-in` concepts as surrogate keys, defined to logically group related data.

Therefore, we should not mine rules which contain relations with instantiations of check-ins or trajectories. We implement that by specifying to the AMIE algorithm which classes should not have instantiations. This feature adds more flexibility to mining instantiated rules, and it is now part of the upstream AMIE 3 code[5] (see Section 6.1).

3.5.3 Injective Mapping

As discussed in subsection 2.4.2, AMIE iteratively builds candidate rules by applying different refinement operators to the previous generation of mined rules. Previous works have noted that AMIE does not force variables mappings to be injective, i.e., two different variables in a rule are allowed to be bound to the same constant. Ebisu and Ichise (2019) argues that injective mappings can be considered to be a better bias for real-world knowledge, and Lajus et al. (2020) discusses that PCA confidence may *underestimate* the likelihood of a prediction in the presence of non-injective mapping.

Next, we present an example[6] which ilustrates the confidence underestimation issue. Consider the triples in Figure 3.8, and the Rule 3.11.

$$hasChild(x, z),\ hasChild(y, z) \Rightarrow marriedTo(x, y) \tag{Rule 3.11}$$

[5] We have contributed to upstream AMIE 3 code a simplified implementation that specifies a set of *relations* that should not have their arguments instantiated. The specification of a set of *classes*, as described in this work, is under review at the moment.

[6] Example taken from the discussion at <https://github.com/lajus/amie/issues/33>

```
marriedTo(Elvis, Priscilla)
hasChild(Elvis, Lisa)
hasChild(Priscilla, Lisa)
```

Figure 3.8 – KB triples used to demonstrate the confidence underestimation issue.

Both injective and non-injective mappings give support of 1, since the set of pairs $\langle x, y\rangle$ that satisfies the rule, $\{\langle Elvis, Priscilla\rangle\}$, has cardinality 1.

If we consider an injective mapping, the rule's confidence is 1/2, since it would predict the following set of tuples $\{\langle Elvis, Priscilla\rangle\ \langle Priscilla, Elvis\rangle\}$. Remember that the rule's (standard) confidence is its support over its number of predictions (subsection 2.4.2).

In a non-injective mapping, such as the case of AMIE's default behavior, the rule's prediction set is $\{\langle Elvis, Priscilla\rangle\ \langle Priscilla, Elvis\rangle, \langle Elvis, Elvis\rangle, \langle Priscilla, Priscilla\rangle\}$, yielding an underestimated confidence of 1/4.

Complementary, here we present an example where non-injective mapping helps eliminate uninteresting rules. Consider the triples in Figure 3.9, and the Rule 3.12.

```
hasChild(Elvis, Lisa)
hasChild(Priscilla, Lisa)
hasBirthday(Elvis, 01/08/1935)
hasBirthday(Priscilla, 05/24/1945)
```

Figure 3.9 – KB triples used to demonstrate the injective rule pruning example.

$$hasChild(x, z),\ hasChild(y, z), hasBirthday(y, b) \Rightarrow hasBirthday(x, b) \qquad \text{(Rule 3.12)}$$

The support of a non-injective mapping will be 2, i.e., the following predicted facts are in KB $\{\langle Elvis, 01/08/1935\rangle\ \langle Priscilla, 05/24/1945\rangle\}$). But when considering an injective mapping, the support will be 0 since there is no binding satisfying $x \neq y$.

Consider again the Rule 3.7 example. We know that *hasTrajectory* is an inverse-function, which means that a trajectory may belong to only one user. Therefore, $u1$ must always be equal to $u2$. Indeed, this rule is mined solely because of non-injective mappings.

Although not available in default settings, AMIE 3 upstream code already has support for mining injective mappings, which we enable in our experiments as described in Chapter 4. We have made major code contributions that significantly speed up AMIE, especially when mining injective rules.

3.5.4 Symmetric-invariant rules

Consider again Rule 3.3 and Rule 3.4. As we previously argued, these rules express the same fact because of *hasFriend*'s symmetric property. Yet, AMIE considers them as two different mined rules.

Dealing with KB data mining in the presence of symmetric relations is a known issue. In special, the link prediction and rule mining tasks can easily experience a *data leakage* scenario in these KBs. As noted in Meilicke et al. (2018) and Akrami et al. (2020), symmetric relations are common on KBs used to train and evaluate these algorithms. Moreover, if not properly handled, they yield overfitted models optimized for the reverse triples (AKRAMI et al., 2020).

Here, we show another issue when mining rules from a KB containing symmetric relations: an exponential combination of rules' predicates that yield equivalent rules but in a non-trivially detectable way. These combinations can easily pollute the mining algorithm's output set.

There is no single or definitive way to deal with these issues. In this work, we present a new but not complete approach to drastically prune and filter the duplicated mined rules[7].

We assume that the user knows a set of symmetric-relations S and that the KB $\mathcal{K}$ that will be mined complies with this set, i.e., $\forall r \in S, r(s,o) \in \mathcal{K} \Leftrightarrow r(o,s) \in \mathcal{K}$. To add symmetric-relation awareness to AMIE, the set S must be specified as one of the algorithm's input.

Our approach is then based on two steps: pruning and filtering. Firstly, our pruning strategy is based on the following definition of a *Symmetric-powerset*:

Definition 3.5.1 (Symmetric-powerset)**.** *Given a rule R and a set of symmetric relations S, we say that the symmetric-powerset $P(R)$ is the set of all rules which can be generated by exchanging one or more $r(s,o) \in R \mid r \in S$ by $r^{-}(o,s)$.*

Example 3.5.1 (Symmetric-powerset example)**.** *Given*

$$R = functional(a, CONST),\ symmetric1(\mathbf{a},\mathbf{b}) \Rightarrow symmetric2(\mathbf{a},\mathbf{b})$$

, and $S = \{symmetric1, symmetric2\}$, the symmetric-powerset $P(R)$ is the set

$$\begin{aligned}\{&functional(a, CONST),\ symmetric1(\mathbf{a},\mathbf{b}) \Rightarrow symmetric2(\mathbf{a},\mathbf{b});\\ &functional(a, CONST),\ symmetric1(\mathbf{a},\mathbf{b}) \Rightarrow symmetric2(\mathbf{b},\mathbf{a});\\ &functional(a, CONST),\ symmetric1(\mathbf{b},\mathbf{a}) \Rightarrow symmetric2(\mathbf{a},\mathbf{b});\\ &functional(a, CONST),\ symmetric1(\mathbf{b},\mathbf{a}) \Rightarrow symmetric2(\mathbf{b},\mathbf{a})\}\end{aligned}$$

Based on the *Symmetric-powerset* of a rule, we can define the *Symmetric-equivalence* of two rules as follows:

[7] For an implementation-side discussion, see <https://github.com/lajus/amie/issues/40>.

Definition 3.5.2 (Symmetric-equivalent)**.** *Given two rules R_1 and R_2, and a set of symmetric relations S, we say that R_1 and R_2 are symmetric-equivalent if and only if R_1 unifies with any $R' \in P(R_2)$.*

Example 3.5.2 (Symmetric-equivalent example)**.** *Given*

$$R_1 = functional(a, CONST),\ symmetric1(a, b) \Rightarrow symmetric2(a, b),$$

$$R_2 = functional(a, CONST),\ symmetric1(b, a) \Rightarrow symmetric2(a, b),$$

$$R_3 = functional(b, CONST),\ symmetric1(a, b) \Rightarrow symmetric2(a, b)$$

, and $S = \{symmetric1, symmetric2\}$, then all three rules are symmetric-equivalent. To prove that R_3 is symmetric-equivalent to R_1, we may swap all symmetric relations in R_3, and then apply the substitution $\sigma = \{a \mapsto b, b \mapsto a\}$ during unification, as shown in the following transformation steps:

$$functional(b, CONST),\ symmetric1(a, b) \Rightarrow symmetric2(a, b)$$
$$functional(b, CONST),\ symmetric1(b, a) \Rightarrow symmetric2(b, a)$$
$$functional(a, CONST),\ symmetric1(a, b) \Rightarrow symmetric2(a, b)$$

At last, we define *Symmetric-aware output* in Definition 3.5.3, which is the set of rules we intend to mine.

Definition 3.5.3 (Symmetric-aware output)**.** *Given a set of mined rules $\mathcal{R}$, we say that $\mathcal{R}$'s symmetric-aware output is a subset $\mathcal{R}' \subseteq \mathcal{R}$ such that there are no pairwise symmetric-equivalent rules in $\mathcal{R}'$.*

Based on these definitions involving rules that contain symmetric-relations, we can propose in Definition 3.5.4 a pruning strategy that greatly reduces the search-space without modifying the set of symmetric-aware output.

Definition 3.5.4 (Symmetric pruning)**.** *Given a rule R, and a set of symmetric relations S, the refining dangling and closing operators are only allowed to add an atom $r(o, s) \mid o > s$, for some partial order operator $>$, if $r \notin S$.*

In order words, Definition 3.5.4 states that if r is a symmetric-relation, the mining operators may add an atom $r(s, o) \in R$, only if $s < o$.

Proof. To prove that Symmetric pruning does not alter the set of symmetric-aware output, we consider $r \in S$ and proceed as follows. If a rule $R = \mathbf{B} \Rightarrow H$ is refined as $R' = r(s, o), \mathbf{B} \Rightarrow H$, then it will also be refined as $R'' = r(o, s) \wedge \mathbf{B} \Rightarrow H$. Since $r \in S \implies r(s, o) \in \mathcal{K} \wedge r(o, s) \in \mathcal{K}$, then R' and R'' will have the same metrics. Consequently, we may remove R'' in favor of R'. Note that by default AMIE already imposes that $H = r(s, o) \mid s \leq o$. □

As we discuss next, this symmetric pruning strategy still produces some symmetric-equivalent rules. We deal with this in a filtering step.

One interesting feature of logical rules is that we can express equivalent rules using multiple combinations of variables identifiers. Consider R_1 and R_3 of Example 3.5.2. We have shown that they are symmetric-equivalent. Even so, they both satisfy the pruning strategy proposed in Definition 3.5.4. A naive approach for building the *symmetric-aware output* would be to check whether two rules are symmetric-equivalent and remove one of them.

Nonetheless, we must consider, in this case, the PCA and its associated confidence metric. As discussed in subsection 2.4.2, PCA assumes that a relation r is an r-attribute of its subject (or object, depending on the relations's functionality). This is especially important when calculating PCA Confidence.

For example, the rules:

$$R_1 = functional(a, CONST),\ symmetric1(a, b) \Rightarrow symmetric2(a, \mathbf{b}),$$

and

$$R_3 = functional(b, CONST),\ symmetric1(a, b) \Rightarrow symmetric2(a, \mathbf{b})$$

would, respectively, yield the PCA queries:

$$Q_1 = functional(a, CONST),\ symmetric1(a, b),\ symmetric2(a, \mathbf{x})$$

and

$$Q_3 = functional(b, CONST),\ symmetric1(a, b),\ symmetric2(a, \mathbf{x})$$

Note that once these queries are built, the symmetric-equivalence is lost, since we cannot change $symmetric2(a, x)$ to $symmetric2(b, x)$[8]. This implies that the PCA Confidence for rules R_1 and R_2 will differ, even though they are symmetric-equivalents.

AMIE runs by firstly mining the set of 1-atom rules, then the set of 2-atom rules, and so on. We propose a filtering step at the end of each mining generation. Indeed, one requirement for symmetric-equivalence is that the rules have the same number of atoms, i.e., they belong to the same generation.

Therefore, at the end of each generation, we have a set of frequent rules which may contain symmetric-equivalences. We proceed by grouping symmetric-equivalent rules using Definition 3.5.5, and then by taking from each group the rule with maximum PCA Confidence, as in Definition 3.5.6.

Definition 3.5.5 (Symmetric-rules partition)**.** *Given a set of mined rules M, we use symmetric-equivalence as a relation to partition M. This yields a partition $SP(M) = \{M_1, M_2, ..., M_k\}$, where all rules within a set M_i are symmetric-equivalent.*

[8] See the Example 3.5.2's last transformation step.

Definition 3.5.6 (Symmetric filtering)**.** *Given a mining generation G_k with all k-atom mined rules, and a set of symmetric relations S, we build a symmetric-aware output $G'_k = \{\, \operatorname{argmax}_{R \in M_i} R : \forall M_i \in SP(G_k) \,\}$.*

Summarizing this work's contributions regarding symmetric-awareness, we have:

1. Added a new bias or pruning strategy (Definition 3.5.4) to the Dangling and Closing operators that prunes out a rule $r(o, s) \wedge \mathbf{B} \Rightarrow \mathbf{H}$ in favor of a rule $r(s, o) \wedge \mathbf{B} \Rightarrow \mathbf{H}$; and
2. Added a new filtering strategy (Definition 3.5.6) that respects PCA Confidence and removes duplicated symmetric-equivalent rules by keeping only the best rule version (the one with highest PCA Confidence).

Note that if a rule contains a head with a symmetric relation, then its absolute support might be doubled since we will be counting both pairs $\langle X, Y \rangle$ and $\langle Y, X \rangle$. We do not tackle this problem, as it would require a deep investigation of special cases, with corresponding changes to AMIE's implementation.

3.5.5 Metrics and rules interpretation

PCA (GALÁRRAGA et al., 2013) and other (in-)completeness-aware metrics (SUCHANEK et al., 2019; TANON et al., 2017) should be carefully explored in the context of semantic trajectories.

Moreover, AMIE's mining process is unaware of the concepts such as *Users* and *Check-ins*, and of how they relate to each other. As a consequence, we cannot easily define domain-specific metrics or mining bias like in Verhein and Chawla (2008), Monreale et al. (2009), and Ghosh and Ghosh (2017).

Also, we note that the rule's interestingness might be better assessed if it is conditioned to some event. For example, consider the Rule 3.13. Note that to calculate the relative support of such rule, we consider all check-ins that are subjects of the *hasTime* relation in our whole dataset, yielding rather low support. Note that if we considered as normalization factor only the *USER 1*'s check-ins, or the check-ins at *Venue 1*, the rule's support would possibly be much higher.

$$\begin{gathered} hasTrajectory(USER\,1, t),\ hasCheckin(t, c), \\ hasVenue(c, Venue\,1) \Rightarrow hasTime(c, Morning) \end{gathered} \qquad \text{(Rule 3.13)}$$

Finally, the data representation is directly related to how we must interpret a rule. Consider Rule 3.14, which states that trajectories which contain a check-in to a specific venue, belong to a specific user. Note that it states a fact regarding *the number of trajectories with visits to this venue*, without considering *the number of visits to this venue*.

$$hasCheckin(t, c),\ hasVenue(c, Venue\,1) \Rightarrow hasTrajectory(USER\,1, t) \qquad \text{(Rule 3.14)}$$

In this work, we do not investigate these different aspects of Semantic Trajectory mining since it would be out of this dissertation's scope of applying a domain-agnostic KB rule mining algorithm. Instead, they should be carefully analyzed in light of a new mining algorithm designed to be aware of the trajectory domain.

3.6 Metarules

AMIE was originally used to mainly mine rules *without* constants. On the other hand, we are interested in rules which describe, for example, specific users, venues, or semantic aspects. Therefore, we mine multiple rules which follow the same pattern but with different instantiated variables. For example, Rule 3.15 and Rule 3.16 express the same concept relations, but with different constants (i.e., *College*/*Shopping*, and *User1*/*User2*).

$$\begin{gathered} hasCheckin(traj,\ checkin),\ hasVenue(checkin,\ venue), \\ hasVenueCategory(venue,\ \boldsymbol{College}) \Rightarrow hasTrajectory(\boldsymbol{User1},\ traj) \end{gathered} \quad \text{(Rule 3.15)}$$

$$\begin{gathered} hasCheckin(traj,\ checkin),\ hasVenue(checkin,\ venue), \\ hasVenueCategory(venue,\ \boldsymbol{Shopping}) \Rightarrow hasTrajectory(\boldsymbol{User2},\ traj) \end{gathered} \quad \text{(Rule 3.16)}$$

Since the search-space strategy of AMIE mines these rules independently, we propose a simple approach to group similar rules based on metarules. A metarule is a rule template which can be used to generate (KAMBER et al., 1997) or to group rules (DJENOURI et al., 2013).

We say that a rule R *complies* with a metarule M if and only if it can be unified with M (KAMBER et al., 1997). Two rules can be unified if there exists a substitution that make them equivalent.

Consider for example the atom $hasVenueCategory(venue,\ College)$ and the atom $hasVenueCategory(a, b)$. They can be unified by applying the substitution $\sigma = \{a \mapsto venue, b \mapsto College\}$. The same idea can be generalized to a whole logical rule.

In this work, we impose that the metarule M is a (Horn) rule with no instantiated atom. Given a set $\mathcal{R}$ of all mined rules, we can construct a set $\mathcal{M}$ of metarules such that every $R \in \mathcal{R}$ complies with one and only one $M \in \mathcal{M}$. For rules R without constants, R is its metarule.

We use $\mathcal{R}_M$ to refer to the subset of mined rules which complies with M. If $R \in \mathcal{R}_M$, we may represent R as a pair $\langle M, T\rangle$, where T is a tuple containing the instantiations in R accordingly to the order that they would appear in M.

We can obtain a set $\mathcal{M}$ from a set $\mathcal{R}$ by applying Algorithm 3.4. The algorithm iteratively builds $\mathcal{M}$ by processing the rules in $\mathcal{R}$. The order in which rules are processed directly influence the $\mathcal{M}$ set that will be obtained, but any such set can be considered equivalent to one another.

The base idea of Algorithm 3.4 is that if a rule R does not already have a metarule in $\mathcal{M}$, it is converted into a metarule and added to $\mathcal{M}$. This conversion can be simply obtained by replacing each rule's variable with a unique variable identifier. Note that the check at Line 4 must also be aware of symmetric-relations, as discussed in Section 3.5.

Algorithm 3.4 Building metarules given a set of mined rules.

```
1: procedure BuildMetarules(Set of rules: $\mathcal{R}$)
2:     $\mathcal{M} \leftarrow$ empty set
3:     for each $R$ in $\mathcal{R}$ do
4:         if $\nexists M \in \mathcal{M} \mid R \in G(M)$ then
5:             $M \leftarrow$ replace $R$'s constants with unique variables
6:             $\mathcal{M} \leftarrow \mathcal{M} \cup \{M\}$
7:     return $\mathcal{M}$
```

Considering Rule 3.15 and Rule 3.16, we can say that they are associated with the metarule Rule 3.17, and represented, respectively, as $\langle \mathit{Rule}\, 3.17, \langle College, User1\rangle\rangle$ and $\langle \mathit{Rule}\, 3.17, \langle Shopping, User2\rangle\rangle$. These tuples correspond to the substitutions $\sigma_1 = \{const1 \mapsto College, const2 \mapsto User1\}$ and $\sigma_2 = \{const1 \mapsto Shopping, const2 \mapsto User2\}$, i.e., $\mathit{Rule}\, 3.15 = \sigma_1(\mathit{Rule}\, 3.17)$ and $\mathit{Rule}\, 3.16 = \sigma_2(\mathit{Rule}\, 3.17)$.

$$\begin{aligned} &hasCheckin(traj,\ checkin),\ hasVenue(checkin,\ venue),\\ &\quad hasVenueCategory(venue,\ \boldsymbol{const1}) \Rightarrow hasTrajectory(\boldsymbol{const2},\ traj) \end{aligned} \tag{Rule 3.17}$$

3.7 Application tasks

As previously discussed, mined rules can communicate hidden mobility patterns and user's preference. The rules mined in this work can take advantage of user-defined relationships between different concepts and multiple semantic aspects. We consider that there are great opportunities to use these patterns in a Knowledge Discovery Framework, with the close cooperation of domain experts.

We also note that the original intent of AMIE is to predict facts that go beyond the set of currently known facts by using logical rules. Other approaches, e.g., based on neural-network embeddings (LIU et al., 2016; FENG et al., 2017; YANG et al., 2019; YANG et al., 2020), can provide state-of-the-art predictive performance for a variety of trajectory-related tasks. Nonetheless, combining both rule mining and neural-network embedding approaches, as will be discussed in Chapter 5, may yield interesting results regarding model interpretability and accuracy.

3.8 Transaction-based and relational-based rules

In this research, we do not offer an empirical comparison of different mining approaches. This motivation is twofold:

1. Current approaches for Semantic Trajectory Association Rules Mining use transaction-based algorithms, which mine rules with smaller expressiveness power when compared to ours.

2. We use AMIE, a state-of-the-art algorithm, as an example to illustrate the benefits of mining Horn rules from Semantic Trajectories while also arguing in favor of a domain-tailored algorithm.

To complement the discussion, we use some rules mined in the literature to contrast with the ones mined with the proposed approach. Consider the following rules, based on those mined in Bogorny et al. (2009), Mousavi et al. (2016), and Khoshahval et al. (2017):

$$\text{if at } Elevado\ das\ Bandeiras \text{ between } [17{:}\,01\text{-}20{:}\,00] \rightarrow \text{next stop is } Avenida\ das\ Americas$$

$$\text{if } Friday \text{ and } EarlyEvening \rightarrow \text{next stop's category is } Entertainment$$

$$\text{if at } Home \text{ and } Evening \rightarrow \text{next stop's category is } Shopping$$

Firstly, note that their antecedents are actually transactional items. This means that a user must define an item to be, for example, a weekday combined with a time period, yielding an item such as $\langle Friday, EarlyEvening\rangle$.

The association rules can also be equivalently represented as Horn Clauses as in Rule 3.18, Rule 3.19, and Rule 3.20. Here, we use a set of relations based on those previously described in this chapter.

$$\begin{split} hasWeekday(x, Friday) \wedge hasTime(x, EarlyEvening) \\ \wedge\, before(x, y) \rightarrow hasVenueCategory(y, Entertainment) \end{split} \tag{Rule 3.18}$$

$$\begin{split} hasVenue(x, Elevado\ das\ Bandeiras) \wedge hasTime(x, [17{:}\,01\text{-}20{:}\,00]) \\ \wedge\, before(x, y) \rightarrow hasVenue(y, Avenida\ das\ Americas) \end{split} \tag{Rule 3.19}$$

$$\begin{split} hasVenuCategory(x, Home) \wedge hasTime(x, Evening) \\ \wedge\, before(x, y) \rightarrow hasVenueCategory(y, Shopping) \end{split} \tag{Rule 3.20}$$

Given this scenario, Horn rules mining represents a step towards more interesting rules that can take advantage of user-defined and domain-specific relations between entities. Indeed, the proposed approach is able to mine Rule 3.18, Rule 3.19, and Rule 3.20, while also being capable to mine arbitrarily relations and semantics represented in the ontology. Examples are Rule 3.6, Rule 3.7, and Rule 3.8.

On the counterpart, it is clear that this flexibility should be compensated with domain-aware biases and metrics. Also, the mined associations cannot be confused with sequential patterns mining.

Chapter 4
EXPERIMENTS

This chapter presents a series of experiments to evaluate our pipeline and investigate what patterns are mined by our proposed approach. We build on the Minimum Working Example experiment and the discussion in Chapter 3.

We make available all source code, experiments setup, results, and analyses at our GitHub repository[1]. Also, Section 6.1 extensively lists technical tools to build this work and experiments. We note, though, that data collected from the Foursquare API is not publicly shared but can still be acquired as discussed in Section 4.1.

4.1 Semantic trajectories

The first step of the pipeline introduced in Chapter 3 encapsulates data acquisition, semantic enrichment, and any pre-processing step required to build Semantic Trajectories. We investigate in this research different datasets from Location-based Social Networks. These datasets are publicly available and are result of previous researches in the Semantic Trajectory community (see subsection 2.3.3).

Data

- Brightkite (CHO et al., 2011)
- Gowalla (CHO et al., 2011)
- Foursquare NYC (Dingqi Yang et al., 2015)
- Foursquare TKY (Dingqi Yang et al., 2015)
- Global Social Foursquare (YANG et al., 2019; YANG et al., 2020)

[1] <https://github.com/falcaopetri/towards-stlarm>

We anticipate that we could not properly mine the datasets Brightkite, Gowalla, and Global Social Foursquare. Their massive size and the domain-unawareness of our mining algorithm yields millions of rules, making it unfeasible and worthless to be analyzed. Instead, we count on the experience of mining Foursquare NYC and Foursquare TKY to guide our analyses and discussion.

Foursquare data can be enriched using venue-related data available in the Foursquare API[2]. In this work, we use the *trajectory-data* code library[3] to access the API and enrich the dataset with the venue's category, price range, and user rating.

Note that there is an anachronism since we retrieve up-to-date data and apply it to check-ins captured many years ago. Although undesirable (LAUBE, 2015), we do not consider it a problem given our research's scope. Some venues are not registered in the Foursquare dataset anymore. We still consider check-ins at these venues and treat the missing semantics as facts not present in the KB.

We enrich the Foursquare NYC dataset with weather data from Wunderground[4], and the Gowalla, BrightKite, and Global Social Foursquare with their respective friendships datasets.

As previously discussed, the main data entities are related to Users, Trajectories, Check-ins, and Venues. Each of these concepts is represented by a unique identifier. For venues, their name is part of the identifier. We represent the following semantic information:

Trajectory-related: each trajectory is associated with the *day of the week*, *month*, and whether it is a Weekday or a Weekend.

Check-in-related: a check-in is associated with a *time of the day* (YANG et al., 2016), which is one of Morning (8:00-12:00), Afternoon (12:00-20:00), or Night (20:00-8:00).

Venue-related: In the case of Gowalla and Brightkite, there are no semantic aspects associated with the venues. For the datasets from Foursquare, the venue has one of the categories: Arts & Entertainment, College & University, Food, Nightlife Spot, Outdoors & Recreation, Professional & Other Places, Residence, Shop & Service, and Travel & Transport. Foursquare NYC and TKY also contain venue's pricing range (*Cheap*, *Moderate*, *Expensive* or *Very Expensive*), and rating, (Low, Medium, or High)[5].

We investigate three different connections between check-ins to represent relations that could not be mined with traditional rule mining. These connections are always calculated between the check-ins within the same trajectory, i.e., we do not connect check-ins that belong to different trajectories.

[2] <https://developer.foursquare.com/docs/places-api/>
[3] <https://github.com/lucaspetry/trajectory-data/>
[4] This data was kindly provided by authors of Petry et al. (2019a).
[5] Venue's rating goes from 4 to 10. We discretize it in equally sized ranges: [4, 6) (Low), [6, 8) (Medium), and [8, 10) (High).

We connect check-ins which occur within a threshold distance using *withinRadius*, and check-ins which occur within a threshold time difference using *withinTimeWindow*. We also establish an order between the check-ins' occurrence using the *before* relation. We apply the thresholds of 2 kilometers distance and 2 hours difference. Note that *withinRadius* and *withinTimeWindow* are symmetric, and *before* is transitive.

Table 4.1 provide a summarized view of the semantic aspects used, excluding the User-to-User and Check-in-to-Check-in relations discussed previously.

Table 4.1 – Semantic Trajectory dimensions description.

Dimension	**Type**	**Range or examples**	**Dataset**
User identifier	Nominal	Unique identifier	All
Place identifier	Nominal	Place name + unique identifier	All
Time	Nominal	{Morning, Afternoon, Night}	All
Day of the week	Nominal	{Mon, Tue, ..., Sun}	All
Weekday	Nominal	{Weekday, Weekend}	All
Month	Nominal	{Jan, Feb, ..., Dec}	All
Weather	Nominal	{Clear, Rain, ..., Snow}[6]	NYC
Rating	Nominal	{Cheap, Moderate, Expensive, Very Expensive}	NYC, TKY
Price Range	Nominal	{Low, Medium, High}	NYC, TKY
Venue Category	Nominal	{Arts & Entertainment, ..., Food, Nightlife Spot}	NYC, TKY, Global

Data filtering

We base our pre-processing steps on previous works that used the Foursquare dataset in the data representation and data mining tasks. Mainly, we filter our data to remove check-ins with too general geo-locations (PETRY et al., 2019a), duplicated data (FERRERO et al., 2020) and to ensure variability in the evaluation (PETRY et al., 2019a).

Following Petry et al. (2019a), we remove check-ins which occur at venues with categories such as *Rivers*, *Cities*, *Neighborhoods*, and *Roads*[7]. The motivation is that the geographic location is unique for each venue, despite the broad area in which the user might have been.

We segment the check-ins into daily trajectories and apply the following sequence of filtering strategies:

1. Remove duplicated check-ins, considering a 10-min threshold (FERRERO et al., 2020);

2. Remove venues with less than 5 check-ins;

[6] The full list of weather conditions is: *Clear, Scattered Clouds, Mostly Cloudy, Overcast, Partly Cloudy, Light Rain, Unknown, Haze, Heavy Rain, Rain, Fog, Mist, Light Snow, Heavy Snow, Snow*.

[7] The full list of excluded categories is: *Bay, Canal, Other Great Outdoors, River, States and Municipalities, City, County, Country, Neighborhood, State, Town, Village, Boat or Ferry, General Travel, Intersection, Moving Target, Road, Taxi, Train Station*.

Table 4.2 – Datasets summary showing the initial and final data size, as well as the impact of each filtering strategy used. We apply the filtering steps in sequential order. The n-th column with percentual value describes the relative size between the original dataset (first data column) and its version after applying the n-th first filters.

(a) Foursquare related datasets. We apply special filtering for broad categories.

		Initial Number	**% after broad category**	**% after de-duplication**	**% after rare venues**	**% after small trajs**	**% after rare users**	**Final number**
NYC	**Users**	1,083	100.00%	100.00%	100.00%	56.60%	13.11%	142
	Trajs.	93,862	97.26%	97.26%	85.15%	5.89%	4.46%	4,184
	Check-ins	227,428	91.51%	90.96%	70.77%	17.74%	14.14%	32,169
	Venues	38,333	95.32%	95.32%	24.48%	16.42%	10.71%	4,104
TKY	**Users**	2,293	100.00%	100.00%	100.00%	61.45%	13.26%	304
	Trajs.	196,713	83.60%	83.60%	71.96%	5.77%	4.09%	8,048
	Check-ins	573,703	64.55%	64.29%	50.94%	14.02%	10.46%	59,982
	Venues	61,858	95.93%	95.93%	22.53%	15.81%	11.89%	7,353
Global	**Users**	114,324	100.00%	100.00%	100.00%	44.47%	6.64%	7,596
	Trajs.	12,173,523	95.89%	95.89%	80.46%	2.56%	1.62%	197,246
	Check-ins	22,809,624	90.19%	89.84%	69.72%	9.08%	6.16%	1,404,976
	Venues	3,820,891	94.72%	94.72%	19.77%	10.00%	6.11%	233,336

(b) Brightkite and Gowalla datasets summary.

		Initial number	**% after de-duplication**	**% after rare venues**	**% after small trajs**	**% after rare users**	**Final number**
Brightkite	**Users**	50,685	100.00%	83.57%	16.64%	4.18%	2,121
	Trajectories	1,683,393	100.00%	77.60%	6.83%	5.88%	99,020
	Check-ins	4,490,260	83.89%	62.11%	18.24%	16.08%	721,901
	Venues	772,705	100.00%	11.04%	5.92%	4.36%	33,720
Gowalla	**Users**	107,068	100.00%	92.43%	26.26%	3.31%	3,541
	Trajectories	2,709,011	100.00%	82.55%	5.55%	3.27%	88,634
	Check-ins	6,442,708	99.61%	72.25%	21.49%	14.09%	907,681
	Venues	1,280,920	100.00%	24.09%	18.87%	15.01%	192,260

3. Remove trajectories with less than 5 check-ins;
4. Remove users with less than 10 trajectories.

We only use friendship data for users who are in our filtered dataset. We note that the filtering process dramatically reduces the number of data points, which ensures that, for example, we only mine rules about the most interesting (frequent) venues. Nonetheless, this also implies that we are applying our approach to much smaller datasets than those found in real-life applications. As we will discuss throughout this chapter, a new domain-tailored algorithm should be developed to handle the domain-specific data volume better.

Table 4.2 summarizes the number of entities in each dataset before and after the filtering process.

Incompleteness assumptions

Based in subsection 3.2.1, we propose a set of incompleteness assumptions to be applied to the datasets. For each assumption, we randomly sample 70% of data to be used in the mining phase.

The assumptions are:

- *Time-related* features: we assume that precise time information may not be available for a given check-in. More specifically, we consider that we know the check-in date, but not its time resolution. We sample 30% of check-ins stratified by user and remove their raw time information. This affects the relations *before*, *withinTimeWindow*, and *hasTime*.
- *Venue-category feature*: we assume that the category of a specific venue may not be available. When available, we stratify by venue category and remove such information from 30% of the venues. This affects the relation *hasVenueCategory*.
- *Venue-related*: we assume that we might not know the exact venue at which a check-in was made. We still know the check-in's geo-location, but not the associated venue. We stratify by venue category (when available) and select 30% of check-ins that will not be connected to a venue. This affects the relation *hasVenue*.
- *User-related*: we assume that we might not know all friends of a user. We randomly remove 30% of all pairs of friendship links. This affects the *hasFriend* relation.

These incompleteness assumptions are applied to each dataset during the rule mining phase and respect the Partial-Completeness Assumption. They are not meant to rigorously evaluate generalization on rule's prediction but to provide an initial discussion about the Standard and PCA Confidences in the context of trajectory data.

Indeed, the initial motivation for applying incompleteness-assumptions was to setup prediction experiments. In this case, the full dataset would be used to evaluate the mined rules' predictions. Nonetheless, we have found the mined rules improper for the prediction task, as many of them predicts too general facts. Still, the incompleteness-assumptions also offers other analyses opportunities, as shown in Section 4.4.

4.2 Domain- and Application-specific representations

Following our data pipeline, we represent each semantic trajectories dataset using STEP and the following relations: *hasTrajectory*, *hasEpisode*, *relatesTo*, *hasFeature*, *hasSemanticDescription*. We extend STEP's classes while populating it with a given dataset, as discussed in Section 3.3.

Table 4.3 – Running time for processing each dataset.

	NYC Foursquare	**TKY Foursquare**	**Global Social Foursquare**	**BrightKite**	**Gowalla**
Data filtering and processing	1.5m	2.5m	2.5h	19m	42m
Build STEP	1.5m	2.5m	1.5h	19m	1.3h
Convert to application-specific	1.8m	1.5m	1.6h	7m	21m
Total	4.8m	6.5m	5.6h	45m	3.4h

We then convert each STEP data to our application-specific representation, which uses the following relations: *hasTrajectory*, *hasCheckin*, *hasVenue*, *hasMonth*, *hasDayOfTheWeek*, *hasWeekday*, *hasVenueCategory*, *hasTime*, *hasPricing*, *hasRating*, *hasWeather*, *before*, *withinRadius*, *withinTimeWindow*. The convertion process is done by the implementation of Algorithm 3.1, Algorithm 3.2 and Algorithm 3.3 without investigating possible parallelization opportunities.

Note that we process each dataset independently. This means that each of them yields a dataset-specific STEP representation and a dataset-specific application representation. Table 4.3 shows the running time for filtering, building the STEP representation, and converting it to our application-specific version.

We present summaries for each pair of STEP and application-specific representations generated by our datasets. To make it more readable, we show here only the summaries for the mined NYC and TKY Foursquare datasets and move the other summaries into Section 6.2. Table 4.4, Table 4.5, Table 6.1, Table 6.2, and Table 6.3 show how our dynamic representation, when compared to STEP:

- uses fewer triples, and especially fewer subjects and objects, to represent data. This is consistent with our intent and approach to removing intermediary FOI entities;
- has more relations as a result of transforming FOI entities into semantic relations;
- has more functional relations since these relationships were previously mixed in a restricted set of STEP relations.

We also show in these tables the summary of full and sampled application-specific representations. In special, Tables 4.4b, 4.5b, 6.1b, 6.2b, and 6.3b presents the PCA-sampled metrics inside parentheses. Values which do not have a corresponding PCA-sampled value were not affected by sampling.

4.3 Rule mining and Metarules

In this section, we run multiple mining experiments. In general, we build each mining experiment as described in subsection 4.3.1, which is consistent with our argumentation from

Table 4.4 – KB summary for NYC Foursquare representations.

(a) KB summary for STEP representation.

Relation	Triples	Func.	Inv. Func.	# subjects	# objects
<hasEpisode>	386,743	0.23	0.08	88,685	32,169
<hasFeature>	172,670	0.60	1.00	104,328	172,670
<relatesTo>	128,209	0.25	0.78	32,169	100,144
<hasSemanticDescription>	83,992	1.00	0.00	83,985	52
<hasTrajectory>	4,184	0.03	1.00	142	4,184
TOTAL	775,798	-	-	309,309	309,219

(b) KB summary for application-specific representation. Values within parentheses are the ones affected by our incompleteness sampling strategy to generate the KB used during the mining phase.

Relation	Triples	Func.	Inv. Func.	# subjects	# objects
<before>	130,842 (63,985)	0.21 (0.29)	0.21 (0.29)	27,985 (18,336)	27,985 (18,336)
<hasCheckin>	32,169	0.13	1.00	4,184	32,169
<hasDayOfTheWeek>	4,184	1.00	0.00	4,184	7
<hasMonth>	4,184	1.00	0.00	4,184	11
<hasVenue>	32,169 (22,518)	1.00	0.13 (0.16)	32,169 (22,518)	4,104 (3,654)
<hasVenueCategory>	4,111 (2,879)	1.00	0.00	4,104 (2,872)	9
<hasPrice>	1,324	1.00	0.00	1,324	4
<hasRating>	1,748	1.00	0.00	1,748	3
<hasTime>	32,169 (22,518)	1.00	0.00	32,169 (22,518)	3
<hasTrajectory>	4,184	0.03	1.00	142	4,184
<hasWeekday>	4,184	1.00	0.00	4,184	2
<hasWeather>	32,088	1.00	0.00	32,088	13
<withinRadius>	124,184	0.23	0.23	27,955	27,955
<withinTimeWindow>	99,548 (48,458)	0.29 (0.38)	0.29 (0.38)	28,561 (18,244)	28,561 (18,244)
TOTAL	507,088 (368,607)	-	-	40,599 (40,044)	40,509 (40,059)

Chapter 3. The only exception is subsection 4.3.2, where we try to mine the STEP ontology instead of our application-specific one.

4.3.1 Setup

Sampled KB: As previously mentioned, we apply a set of sample strategies to manually inject incompleteness in our KBs.

Bias: Unless specified, we mine rules with *injective mappings*, using our proposed *symmetric-aware pruning and filtering strategies*, and *disallowing instantiations of* `Trajectory` *and* `Check-in` *instances* (Section 3.5).

We always mine 4-atom rules. We empirically find this number to be feasible to be mined given a reasonable time and allow a large number of candidate patterns.

We also always use a recursive limit of 2, which means that a rule can use a relation r at most 2 times. AMIE's default value is 3, but since we mine 4-atom rules, this would uninterestingly allow 3 out of 4 atoms to have the same relation.

Table 4.5 – KB summary for TKY Foursquare representations.

(a) KB summary for STEP representation.

Relation	**Triples**	**Func.**	**Inv. Func.**	**# subjects**	**# objects**
<hasEpisode>	767,038	0.22	0.08	168,563	59,982
<hasFeature>	265,178	0.51	1.00	134,950	265,178
<relatesTo>	179,531	0.33	0.71	59,982	126,902
<hasSemanticDescription>	96,634	1.00	0.00	96,615	39
<hasTrajectory>	8,048	0.04	1.00	304	8,048
TOTAL	1,316,429	-	-	460,414	460,149

(b) KB summary for application-specific representation. Values within parentheses are the ones affected by our incompleteness sampling strategy to generate the KB used during the mining phase.

Relation	**Triples**	**Func.**	**Inv. Func.**	**# subjects**	**# objects**
<before>	235,164 (114,839)	0.22 (0.30)	0.22 (0.30)	51,934 (33,943)	51,934 (33,943)
<hasCheckin>	59,982	0.13	1.00	8,048	59,982
<hasDayOfTheWeek>	8,048	1.00	0.00	8,048	7
<hasMonth>	8,048	1.00	0.00	8,048	11
<hasVenue>	59,982 (41,987)	1.00	0.12 (0.16)	59,982 (41,987)	7,353 (6,563)
<hasVenueCategory>	7,372 (5,167)	1.00	0.00	7,353 (5,149)	9
<hasPrice>	1,344	1.00	0.00	1,344	4
<hasRating>	3,792	1.00	0.00	3,792	3
<hasTime>	59,982 (41,987)	1.00	0.00	59,982 (41,987)	3
<hasTrajectory>	8,048	0.04	1.00	304	8,048
<hasWeekday>	8,048	1.00	0.00	8,048	2
<withinRadius>	244,496	0.22	0.22	53,031	53,031
<withinTimeWindow>	227,396 (111,334)	0.24 (0.33)	0.24 (0.33)	55,550 (36,211)	55,550 (36,211)
TOTAL	931,702 (657,120)	-	-	75,687 (74,142)	75,422 (74,632)

Metrics and thresholds: We use AMIE's default parameters. This means pruning based on a minimum Head Coverage of 0.01, and filtering based on a minimum PCA Confidence of 0.1. These values should be properly tuned by a domain-expert and by considering the dataset characteristics. Nonetheless, we do not change them as they cannot fix the inherently inappropriate mining bias, as discussed in Chapter 3 and Section 5.1.

RDF data: For each experiment, the populated ontology with all materialized facts is exported as RDF triples in the N-triples format and are used as input to AMIE. Triples that state facts about the ontology schema are removed since they are not used. The triples input is similar to Figure 3.7.

To enable symmetric-awareness, we use a separate file to state the schema facts which define symmetric relations. The same schema file is used to specify the domain and range of each relation in the ontology. This information can then be used to prune and filter symmetric-equivalent rules and to avoid the instantiation of specific classes, as discussed in Section 3.5. In our experiments, we use the schema file shown in Figure 4.1.

```
<withinRadius> rdf:type owl:symmetric .
<hasFriend> rdf:type owl:symmetric .
<withinTimeWindow> rdf:type owl:symmetric .

<withinTimeWindow> rdfs:range <Checkin> .
<withinTimeWindow> rdfs:domain <Checkin> .

<withinRadius> rdfs:range <Checkin> .
<withinRadius> rdfs:domain <Checkin> .

<hasWeather> rdfs:range <VenueCategory> .
<hasWeather> rdfs:domain <Checkin> .

<hasWeekday> rdfs:range <VenueCategory> .
<hasWeekday> rdfs:domain <Trajectory> .

<hasTrajectory> rdfs:range <Trajectory> .
<hasTrajectory> rdfs:domain <User> .

<hasTime> rdfs:range <VenueCategory> .
<hasTime> rdfs:domain <Checkin> .


<hasRating> rdfs:range <VenueCategory> .
<hasRating> rdfs:domain <Venue> .

<hasPrice> rdfs:range <VenueCategory> .
<hasPrice> rdfs:domain <Venue> .

<hasVenueCategory> rdfs:range <VenueCategory> .
<hasVenueCategory> rdfs:domain <Venue> .

<hasVenue> rdfs:range <Venue> .
<hasVenue> rdfs:domain <Checkin> .

<hasMonth> rdfs:range <VenueCategory> .
<hasMonth> rdfs:domain <Trajectory> .

<hasDayOfTheWeek> rdfs:range <VenueCategory> .
<hasDayOfTheWeek> rdfs:domain <Trajectory> .

<hasCheckin> rdfs:range <Checkin> .
<hasCheckin> rdfs:domain <Trajectory> .

<before> rdfs:range <Checkin> .
<before> rdfs:domain <Checkin> .
```

Figure 4.1 – Schema file used in our symmetric-aware experiments. Unlike those in Figure 3.7, the facts here are shown in the N-triples format, the input format used by AMIE.

Hardware setup: We executed our experiments in a machine with Intel® Core™ i9-7900X CPU @ 3.30GHz, 126 GiB system memory, and 20 cores.

4.3.2 Mining rules from STEP

In Chapter 3, we have argued that mining Horn rules from STEP is a challenging task since the rules can only use a limited set of relations. Instead, we proposed an application-specific representation that is both automatically derived from STEP and capable of yielding interesting rules.

In this experiment, we investigate mining the NYC Foursquare dataset considering the STEP ontology representation. We run AMIE as previously described with little configuration changes. Besides using the STEP representation as input to the algorithm, we use the schema triples shown in Figure 4.2. We mine rules with injective-mapping and ignoring the instantiation of `Episode` and `Trajectory` entities. Symmetric-relation-awareness is not explored since there are no symmetric relations in STEP.

```
<hasTrajectory> rdfs:domain <Agent> .
<hasTrajectory> rdfs:range <Trajectory> .
<relatesTo> rdfs:domain <Episode> .
<relatesTo> rdfs:range <ContextualElement> .
<hasEpisode> rdfs:domain <FeatureOfInterest> .
<hasEpisode> rdfs:range <Episode> .
<hasFeature> rdfs:domain <SpatiotemporalElement> .
<hasFeature> rdfs:range <FeatureOfInterest> .
<hasSemanticDescription> rdfs:domain <FeatureOfInterest>
<hasSemanticDescription> rdfs:range <SemanticDescription> .
```

Figure 4.2 – Schema file used in our experiment on mining the STEP ontology. Unlike Figure 3.7, the facts here are shown in the N-triples format, the input format used by AMIE.

The result can be seen in Table 4.6. A total of 50 rules were mined and generated 3 different metarules. We use the tuple notation introduced in Section 3.6 to denote a rule given its associated metarule.

Table 4.6 – Summary of the 50 rules mined by AMIE from the STEP NYC Foursquare representation. The rules can be grouped into 3 metarules, for which we show instantiations examples following the notation from Section 3.6. *HC* stands for Head Coverage.

ID	Metarule	HC range	Std Conf. range	PCA Conf. range	# rules	Rule tuple	HC	Std Conf.	PCA Conf.
Rule 4.1	$hasFeature(t, f1), hasFeature(t, f2), hasSemanticDescription(f2, const1) \Rightarrow hasSemanticDescription(f1, const2)$	[0.0100, 0.0263]	[0.09, 0.27]	[0.13, 0.40]	2	$\langle May, Weekday \rangle$	0.0103	0.27	0.40
						$\langle Weekday, May \rangle$	0.0103	0.09	0.13
Rule 4.2	$hasFeature(t, f), hasEpisode(f, e), relatesTo(e, const1) \Rightarrow hasTrajectory(const2, t)$	[0.0100, 0.0263]	[0.07, 0.16]	[0.75, 1.00]	47	$\langle POI\ SanMarcoPizzeria\_4b12f, User\ 293 \rangle$	0.0143	0.09	1.00
						$\langle POI\ SettepaniBakery\_429ba, User\ 293 \rangle$	0.0134	0.09	1.00
						$\langle POI\ Mcdonalds\_4c9f0, User\ 354 \rangle$	0.0115	0.08	0.91
Rule 4.3	$hasFeature(c, f1), hasEpisode(f1, e), hasFeature(c, f2) \Rightarrow hasEpisode(f2, e)$	[0.6517, 0.6517]	[0.43, 0.43]	[0.51, 0.51]	1	$\langle \, \rangle$	0.6517	0.43	0.51

Rule 4.1 states that, given two FOI's $f1$ and $f2$ from t and a semantic description of $f2$, then we can predict a semantic description of $f1$. Note that although t might be a `Trajectory` or a `ContextualElement`, the instantiated rules shows that in this case t is always a trajectory. Therefore, the rule predicts a trajectory's semantics based on another semantic. In the case of the rule $\langle Rule\, 4.1, \langle May, Weekday \rangle \rangle$, we are predicting that trajectories in May are usually $weekday$ trajectories.

Rule 4.2 has 47 different rules associated to it. By trying to interpret it, we can see that it is actually predicting a user based on the visit venue. For example, $\langle Rule\, 4.2, \langle POI\ SanMarcoPizzeria\_4b12f, User\ 293 \rangle \rangle$ is associating the visit to the venue called *San Marco Pizzeria* to the user with identifier 293 with 100% (PCA) confidence.

We can note the huge boost on these rules' PCA Confidence when compared to their Std Confidence. Since the rule's head contains the *hasTrajectory* relation, the PCA uses as the rule's counter-examples only the trajectories from other users who have also visited that specific venue. On the other hand, Std Confidence allows t to bind with many `ContextualElement's`, which greatly increases the number of counter-examples.

Finally, Rule 4.3 is a metarule of itself since it does not contain any constant. At first sight, it might appear that $f1$ and $f2$ are always bound to the same value, which would contradict our injective-mining bias. We can infer by the ontology schema that c is always a *ContextualRelations* instance, which we use to group connections between *Episode*'*s*, such as *withinRadius*, *withinTimeWindow* and *before*. Therefore, if an *Episode* appears in the *ContextualRelations*'s *withinTimeWindow* list, then it is also expected for example to happen in its *before* list.

In general, these metarules hint at the importance of mining typed rules, i.e., rules that specify their variables' type. We can see that some semantic relationships (Rule 4.2) can be mined, but they are very limited given the approach to encode semantics using `FeatureOfInterest` entities.

4.3.3 Global, Brightkite, and Gowalla

We let AMIE mine each of the Global Social Foursquare, Brightkite, and Gowalla datasets for at least 3 full days. Even after this considerable time slot, their mining phase was still refining 2-atom rules into 3-atom rules. Due to shared hardware constraints, we proceeded to shut down the processes once the allocated time was reached.

Unfortunately, AMIE does not have a straightforward approach to set a timeout parameter or even to collect partial mining results[8]. What we can recover from AMIE's run is a set of long-running queries and the time spent on processing them.

For example, Rule 4.4 takes 1.17 hours to run, and Equation 4.1 takes 2.77 hours. The first pattern is an example of the dangling-operator query, which tries to find the suitable set of

[8] This feature is under discussion at <https://github.com/lajus/amie/issues/54>.

Table 4.7 – Summary of mined rules and metarules, and other performance metrics.

	NYC	TKY
# rules	70,599	30,985
# metarules	1,383	943
Runtime	4h30	7h40
Max. memory	7 GB	10 GB

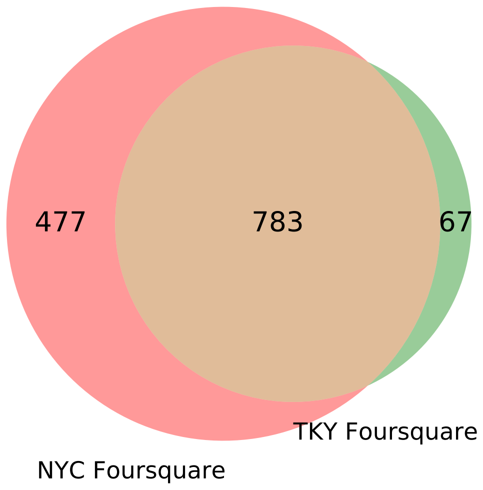


Figure 4.3 – Venn diagram of metarules in NYC and TKY datasets. The values show the cardinality of each subset.

relations to bind with x. The second pattern is an example of a PCA query (the head relation $hasCheckin$ contains an unbound variable x) with injective mapping (the variables a and b cannot bound with the same constant). There are many possible bindings in both cases since almost any check-in pair could be a valid binding to a and b (the head variables).

$$hasDayOfTheWeek(i,b),\ hasTrajectoryCategory(a,Weekday),\\ x(i,n) \Rightarrow hasDayOfTheWeek(a,b) \qquad \text{(Rule 4.4)}$$

$$hasTime(b,Afternoon),\ hasDayOfTheWeek(a,Wednesday),\\ a \neq b,\ hasCheckin(x,b) \qquad \text{(Equation 4.1)}$$

Even though mining these different datasets would be interesting, we consider it to have little impact on our discussion.

4.3.4 NYC and TKY Foursquare

From the NYC and TKY Foursquare datasets, we mine a total of 101,584 rules as shown in Table 4.7. The table also shows AMIE's running time for each dataset and the maximum memory used by the program during its execution.

Complementary, we show in Figure 4.3 that there is a large intersection between the metarules mined in both datasets. As can be expected, the number of unique metarules in NYC Foursquare is the largest one since the dataset has one relation more than TKY Foursquare (*hasWeather*).

4.4 Rules analyses

In this section, we investigate the rules mined from the NYC and TKY Foursquare datasets. We begin by noting the large number of rules mined. Even considering the aggregated mined rules, the two experiments yielded a total of 1,323 different metarules (as previously shown in Figure 4.3).

Next, we explore both quantitatively and qualitatively different types of rules.

4.4.1 Graphical analyses

For consistency, we take special care to make the NYC and TKY graphics compatible since their set of relations differ in one. When necessary, we omit or leave a blank space in our graphs for the *hasWeather* relation. We also maintain the same color mapping across both datasets and all graphics, such that a relation is always represented with the same color.

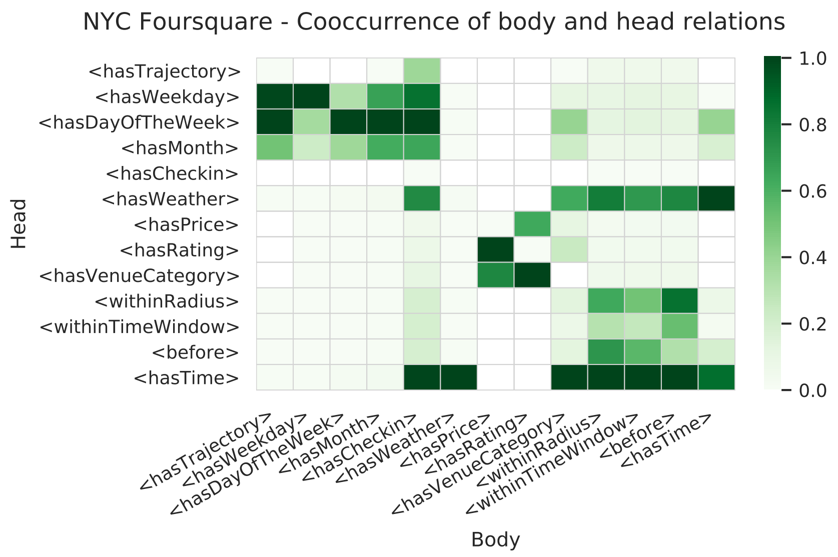
Given a rule $\mathbf{B} \Rightarrow h(x, y)$, Figure 4.4 shows a simple analysis considering how the relations in B are related to the predicted relation h. We do this by counting the frequency in which $r \in \mathbf{B}$ occurs with the relation h. We normalize the rows in Figure 4.4 to tell for each head relation the most frequent body relations associated. This is not a meta-association or a formal correlation analysis, but it is intended to raise some initial semantic observations.

We sort the relations by the following criteria: relations related to the trajectory (*hasTrajectory*, *hasWeekday*, *hasDayOfTheWeek*, *hasMonth*, *hasCheckin*), relations related to a check-in (*hasWeather*), relations related to a venue (*hasPrice*, *hasRating*, *hasVenueCategory*), spatial relations (*withinRadius*), and temporal relations (*withinTimeWindow*, *before*, *hasTime*).

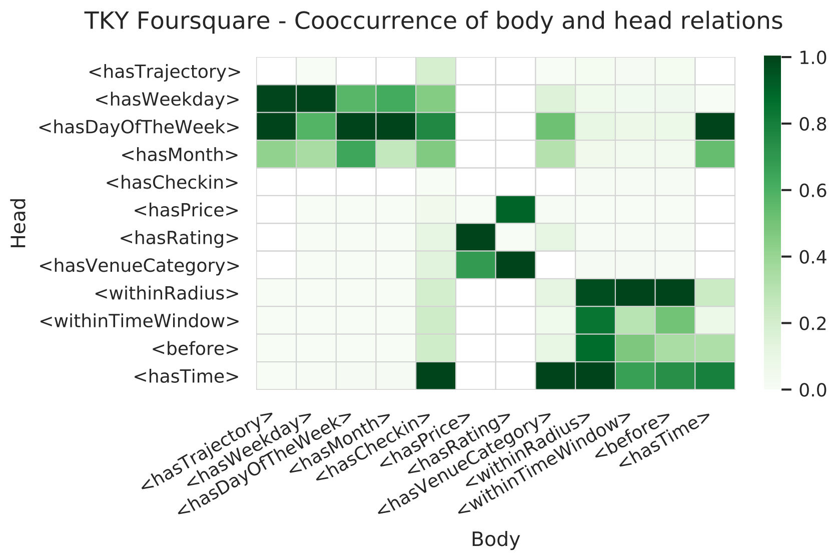
Figures 4.4a and 4.4b shows some interesting coocurrences, highlighted by our relations sorting criteria. For example, we can see at least two very correlated set of relations: { *hasTrajectory*, *hasWeekday*, *hasDayOfTheWeek*, *hasMonth* }, and { *hasVenueCategory*, *withinRadius*, *withinTimeWindow*, *before*, *hasTime* }. We can also see some connetions between the relations *hasPrice*, *hasRating*, and *hasVenueCategory*.

Next, Figure 4.5 and Figure 4.6 show how the number of rules and metarules varies given different confidence thresholds. There are two important ways to explore these graphics. Firstly, in a columnar view we can compare the number of rules and metarules. For example, consider the first column. We can note that varying the Std Confidence thresholds yields similar curves for the number of rules and the number of metarules. The same thing can be noted when considering the second column, which varies the PCA confidence threshold.

The second way to explore the graphic is to consider it row-wise. We can see that some specific head relations curves are altered when we compare Std and PCA confidences. More specifically, we can see that the curves for *hasRating* and *hasPrice* are elongated when we consider the PCA Confidence. This is consistent with the characteristics of these relations: they



(a) NYC Foursquare data.



(b) NYC Foursquare data.

Figure 4.4 – Coocurrence of body and head relations. The heatmap is normalized in each row.

are functional relations with many missing values. Therefore, assuming the PCA in these cases allows AMIE to model a more optimistic confidence value.

We can also see some (meta)rules head relations with little or no variation as we change the confidence thresholds. This hints at the existence of very generic rules, which contain extensive coverage or represent particular data associations.

Although there are some small variations between Figure 4.5 and Figure 4.6, we consider that they mostly demonstrate the same behavior.

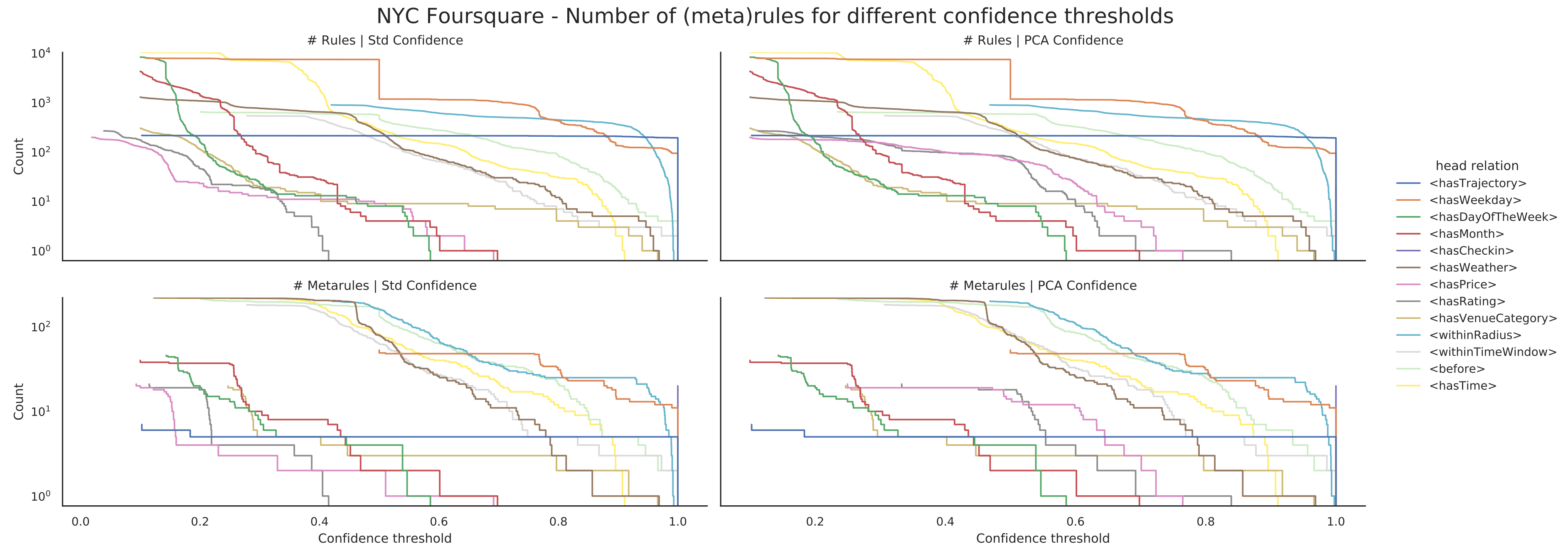


Figure 4.5 – The number of *rules* (first row) and *metarules* (second row) given different *Std confidence* (first column) and *PCA confidence* (second column) thresholds for the **NYC Foursquare** dataset. By experiment design, we only mine rules with at least 0.1 PCA Confidence. The Y axes are in log scale, and they are shared between the plots in each row.

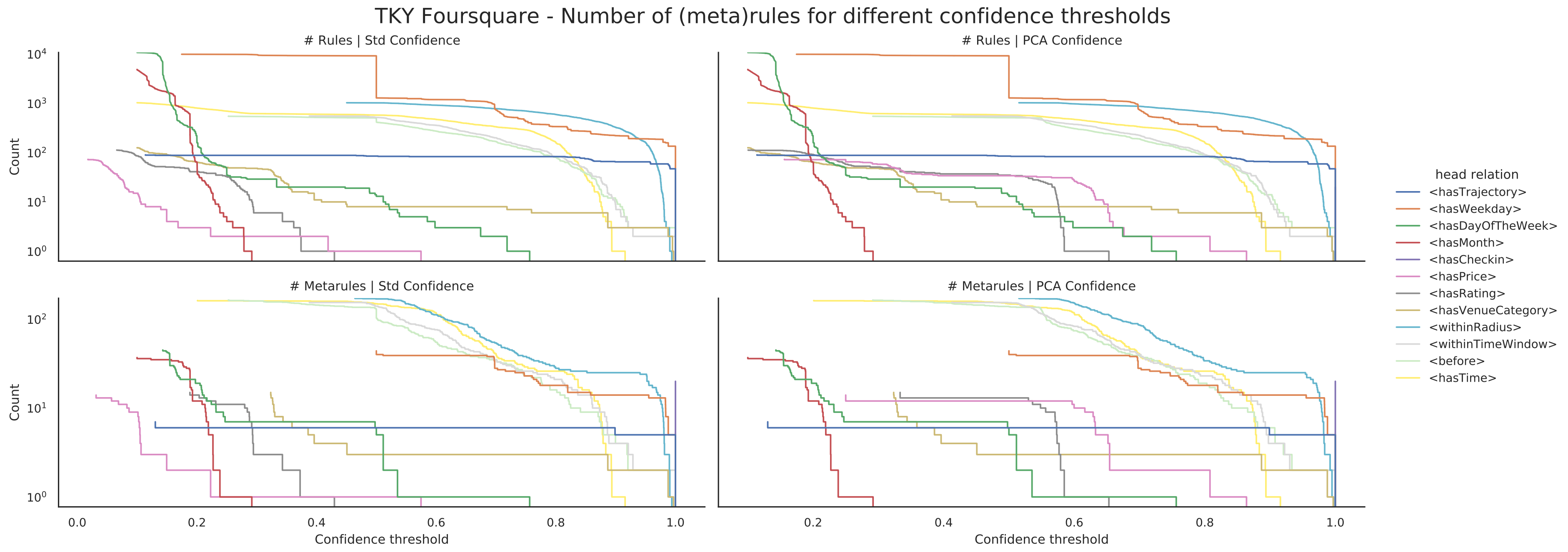


Figure 4.6 – The number of *rules* (first row) and *metarules* (second row) given different *Std confidence* (first column) and *PCA confidence* (second column) thresholds for the **TKY Foursquare** dataset. By experiment design, we only mine rules with at least 0.1 PCA Confidence. The *Y* axes are in log scale, and they are shared between the plots in each row.

Figure 4.7 and Figure 4.8 show how the number of rules and metarules varies given different absolute and relative support thresholds. These figures present interesting observations when we consider them column-wise.

By considering the positive examples (i.e., absolute support) column, we can see different order of magnitudes for different head relations groups. The relations *withinRadius* , *withinTimeWindow*, and *before* have a very high support given their characteristic of connecting many check-ins.

We also have a group of curves related to the *hasCheckin*, *hasWeather*, and *hasTime* relations. This makes sense since the number of different target entities bounds the absolute support. In this case, these relations are bounded by the number of check-ins. Next, we can see in the graphic the set of relations bounded by the number of venues and trajectories.

Moving to the second column, we see how the Head Coverage compensates the different bounds imposed by absolute support. Even so, we can also see other sets of similar curves. For example, *hasPrice* and *hasRating* have very similar curves, while *hasTrajectory* and *hasVenueCategory* (meta)rules diminishes rapidly with the increase of the Head Coverage threshold.

Figure 4.9 complements the Head Coverage discussion, since it shows that the *hasTrajectory* and *hasVenueCategory* are the only two relations that predicts only constants. As defined in subsection 2.4.2, the Head Coverage does not consider if the head atom predicts a constant (e.g., $hasTrajectory(User, t)$: it will always normalize by the number of distinct relation facts (e.g., $hasTrajectory(u, t)$)[9]. Therefore, a rule that predicts a constant still gets penalized by the Head Coverage.

[9] See a possible alternative definition/implementation in <https://github.com/lajus/amie/issues/35>.

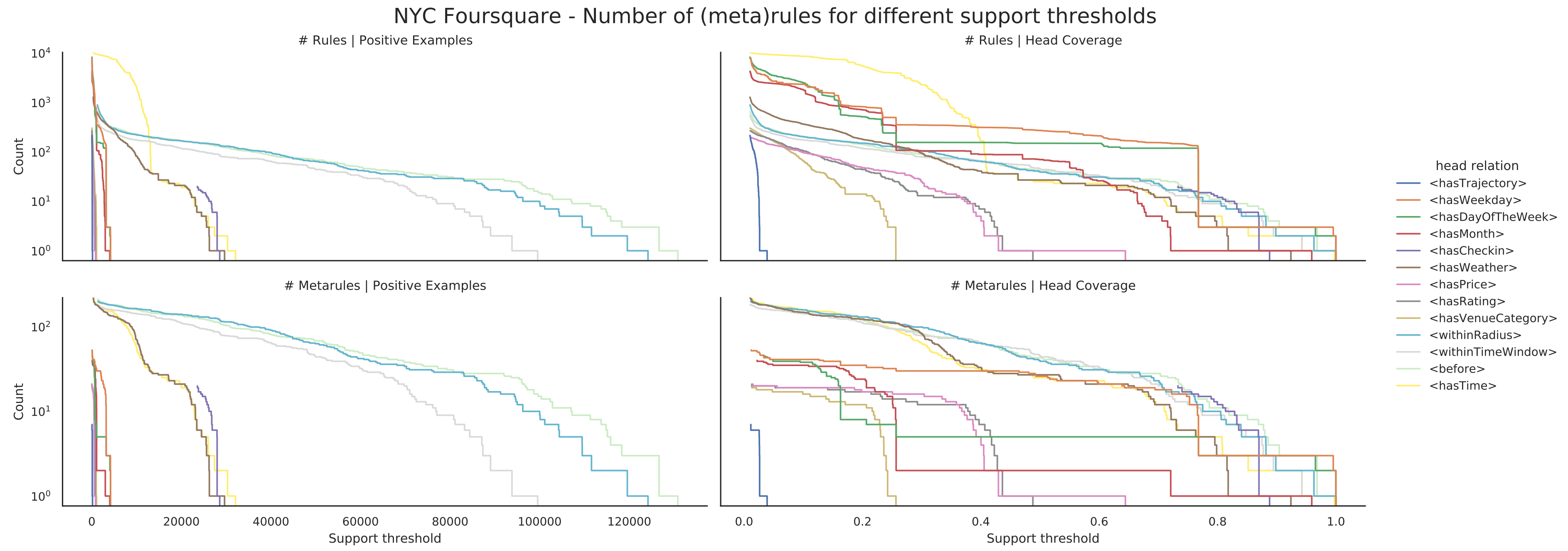


Figure 4.7 – Number of rules (first row) and metarules (second row) given different absolute support (first column) and Head Coverage (second column) thresholds for the **NYC Foursquare** dataset. Remember that by experiment design, we only mine rules with at least 0.01 Head Coverage. Note that the Y axes are in log scale, and that they are shared between the plots in each row.

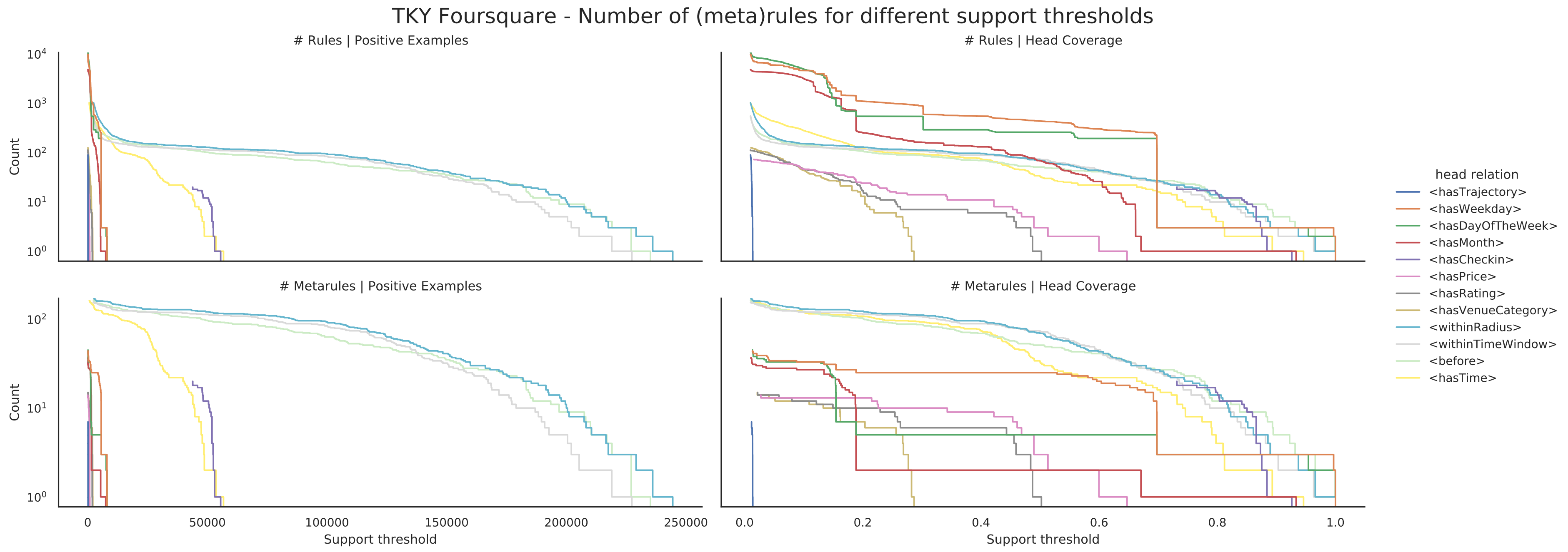


Figure 4.8 – Number of rules (first row) and metarules (second row) given different absolute support (first column) and Head Coverage (second column) thresholds for the **TKY Foursquare** dataset. Remember that by experiment design, we only mine rules with at least 0.01 Head Coverage. Note that the Y axes are in log scale, and that they are shared between the plots in each row.

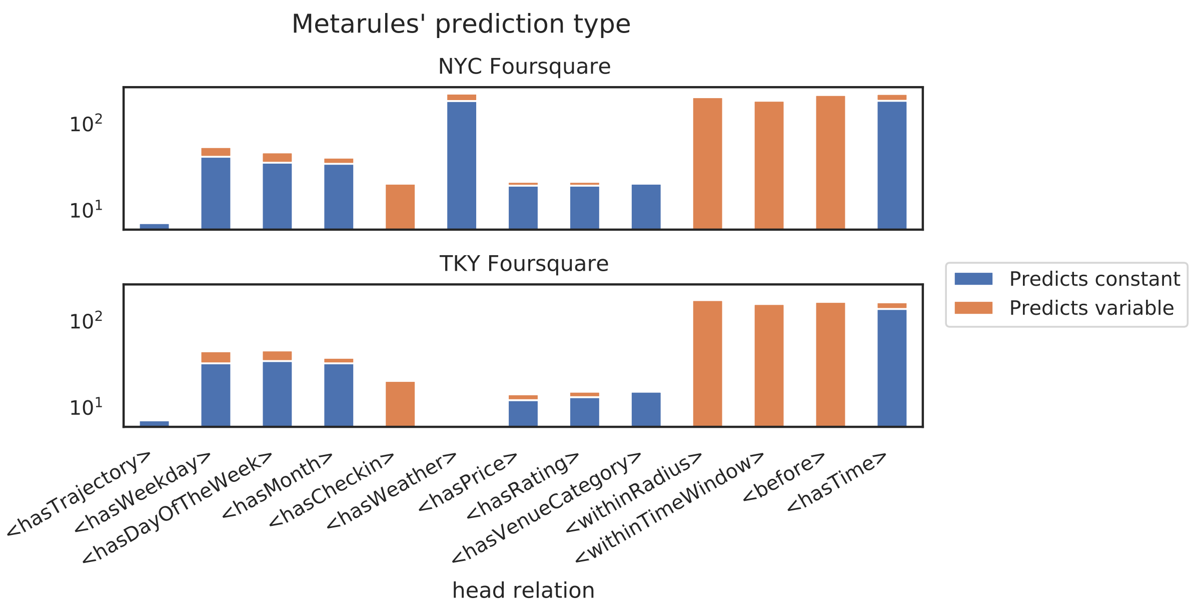


Figure 4.9 – Number of metarules for each head relation considering whether the metarule predicts a constant or a variable.

Finally, Figure 4.10 shows another visualization indicating which relations the PCA affects the most. More specifically, we can see how *hasPrice* and *hasRating* have consistent boosts in the estimated confidence.

4.4.2 Qualitative analyses

Without a proper search bias or domain constraints, we mine many rules using multiple *before*, *withinRadius*, and *withinTimeWindow* relations. Since these relations connect a large

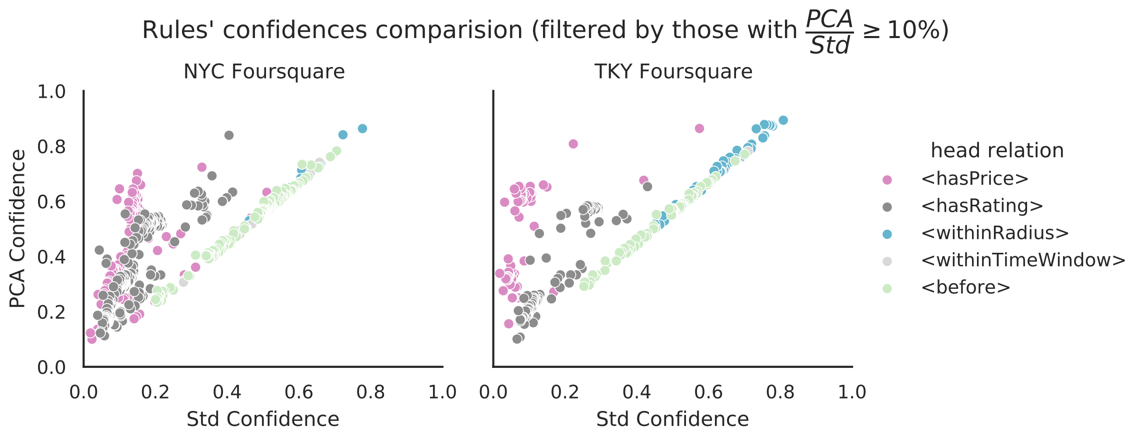


Figure 4.10 – Rules' PCA vs Std confidences. For clarity, we select only those rules with a confidence difference of at least 10%.

Table 4.8 – Count of metarules per number of target relations occurrence. We consider the distinct metarules mined in NYC and TKY datasets.

# *before*, *withinRadius*, and *withinTimeWindow* relations	**# metarules**
0	163
1	225
2	341
3	433
4	165

amount of check-ins pairs, they commonly yield generic rules with no semantics information, such as in Rule 4.5.

$$\begin{gathered} before(a,b),\ before(b,h),\ withinTimeWindow(a,h) \\ \Rightarrow withinTimeWindow(a,b) \end{gathered} \tag{Rule 4.5}$$

We count how many of these relations appear in each metarule and summarize the result in Table 4.8. We can see that there are a large number of metarules that just combine these relations. Even rules with 2 of them, such as Rule 4.6, contain uninteresting associations.

$$\begin{gathered} before(a,d),\ hasVenue(h,TheBushHotTub\_4f22), \\ withinRadius(d,h) \Rightarrow hasTime(a,Afternoon) \end{gathered} \tag{Rule 4.6}$$

To proceed with our analyses, we ignore all rules with 2 or more occurrences of the discussed relations. We then manually analyze the remaining 388 metarules. We look for:

- patterns that are clearly spurious correlations;
- patterns that could (and should) be pruned using domain knowledge or some custom bias;
- patterns that represent interesting connections;
- interesting patterns that could only be mined by our relational-based approach.

We select patterns representing our domain-unaware logical rule mining process' strengths and weaknesses and group them into different categories. Table 4.9 shows pattern examples that capture some rules used to describe the dataset construction.

1. Rule 4.7 describes that two trajectories with the same *day of the week* always have the same *weekday*. This and other similar patterns show the inter-dependencies of different semantic aspects on our semantic processing construction.
2. Rule 4.8 describes a domain rule introduced in our data: check-ins connected through the relation *before* always belong to the same trajectory. We mine similar rules for the relations *withinTimeWindow* and *withinRadius*.

3. Rule 4.9 is similar to Rule 4.8, but with the arguments of the *before* relation reversed. This shows that although the *before* relation is not symmetric, it may still yield uninteresting pairs of rules. Note that this pair is not symmetric-equivalent (Definition 3.5.2).

Note that all rules in Table 4.9 have a very high Head Coverage and confidence scores equal to 1. This shows that we have mined rules with wide coverage exploring the underlying data semantics definition to make them (almost) always correct.

Next, we explore in Table 4.10 some rules that express spurious data correlations.

1. Rule 4.10 shows that there are pairs of users whose trajectories are almost always with the same weekday. This hints at a bias in the dataset towards these users checking-in more commonly on either weekdays or weekends.

2. Rule 4.11 relates a venue's category and the check-in's weather condition to the venue's pricing. Of course, this is a spurious correlation since pricing has a fixed value with no temporal dimension. If AMIE has outputted this rule, it has greater confidence than its alternative without the weather condition.

Table 4.11 shows a set of rules that offers an interesting opportunity for future investigations: to check how mutually exclusive values differ from the expected apriori confidences scores. Take as an example the rule Rule 4.12, which shows an association between a specific user's check-in time and whether it is a weekday or weekend. Although this might be due to data biases and spurious correlations, we consider it interesting to compare similar rules that use, for example, mutually exclusive constants.

For example, consider that 90% of the user's check-ins are during workdays, and 10% are during weekends. If we mine a rule that changes this apriori confidence by using a semantic aspect, we can explain and model the user's behavior during weekends. Similar hypotheses could be created by considering the other examples in Rule 4.13 and Rule 4.14.

Table 4.9 – Example of (meta)rules that describe data semantics.

	Metarule	Rule	Head Coverage	Std Confidence	PCA Confidence
Rule 4.7	$hasDayOfTheWeek(a,\ f),\ hasDayOfTheWeek(i,\ f),\ hasWeekday(i,\ b) \Rightarrow hasWeekday(a,\ b)$	$\langle\,\rangle$	1.0000	1.0	1.0
Rule 4.8	$before(b,\ f),\ hasCheckin(a,\ f) \Rightarrow hasCheckin(a,\ b)$	$\langle\,\rangle$	0.8699	1.0	1.0
Rule 4.9	$before(f,\ b),\ hasCheckin(a,\ f) \Rightarrow hasCheckin(a,\ b)$	$\langle\,\rangle$	0.8699	1.0	1.0

Table 4.10 – Example of (meta)rules that describe spurious correlations.

	Metarule	Rule	Head Coverage	Std Confidence	PCA Confidence
Rule 4.10	$hasTrajectory(CONST\_0,\ a),\ hasTrajectory(CONST\_1,\ i),\ hasWeekday(i,\ b) \Rightarrow hasWeekday(a,\ b)$	$\langle User\_384, User\_820\rangle$	0.0165	0.90	0.90
		$\langle User\_384, User\_262\rangle$	0.0165	0.90	0.90
Rule 4.11	$hasVenue(e,\ a),\ hasVenueCategory(a,\ CONST\_0),\ hasWeather(e,\ CONST\_1) \Rightarrow hasPrice(a,\ CONST\_2)$	$\langle Food, Haze, Cheap\rangle$	0.0491	0.69	0.76
		$\langle Food, HeavyRain, Cheap\rangle$	0.0136	0.64	0.72

Table 4.11 – Example of (meta)rules that describe possibly interesting patterns.

	Metarule	Rule	Head Coverage	Std Confidence	PCA Confidence
Rule 4.12	$hasCheckin(a,\ f),\ hasTime(f,\ CONST\_0),\ hasTrajectory(CONST\_1,\ a) \Rightarrow hasWeekday(a,\ CONST\_2)$	$\langle Afternoon, User\_384, Weekday\rangle$	0.0120	0.96	0.96
		$\langle Morning, User\_384, Weekday\rangle$	0.0134	0.95	0.95
Rule 4.13	$hasCheckin(g,\ c),\ hasDayOfTheWeek(g,\ CONST\_0),\ hasVenue(c,\ a) \Rightarrow hasVenueCategory(a,\ CONST\_1)$	$\langle Sunday, Food\rangle$	0.0873	0.26	0.26
		$\langle Saturday, Food\rangle$	0.1080	0.25	0.25
Rule 4.14	$hasCheckin(g,\ c),\ hasVenue(c,\ a),\ hasWeekday(g,\ CONST\_0) \Rightarrow hasPrice(a,\ CONST\_1)$	$\langle Weekday, Cheap\rangle$	0.4056	0.15	0.52
		$\langle Weekend, Cheap\rangle$	0.2440	0.14	0.45

Finally, we show in Table 4.12 a set of patterns that we consider interesting. It is important to highlight that rules Rule 4.16, Rule 4.17, Rule 4.18, Rule 4.19, and Rule 4.20 contain relations that can be mined only with our proposed approach.

- Rule 4.15 shows how we can use venues to predict the associated users. It is similar to the rule mined in out STEP experiment (Rule 4.2), but uses one relation less due to our application-specific representation.
- Rule 4.16 shows the average frequency in which two check-ins from the same trajectory are within a radius of 2 km (the threshold we used to define the *withinRadius* relation in subsection 4.3.1).
- Rule 4.17, Rule 4.18, and Rule 4.19 show different semantics that we can add to Rule 4.15 in order to improve the rule's confidence.
- Rule 4.20 describes that when two different check-ins are registered in the same venue within a time window, then the venue's category is *TravelTransport*. This hints at a user's behavior of checking-in multiple times at certain conditions.

Table 4.12 – Example of (meta)rules that describe interesting patterns.

	Metarule	**Rule**	**Head Coverage**	**Std Confidence**	**PCA Confidence**
Rule 4.15	$hasCheckin(b,\ d),\ hasVenue(d,\ CONST\_0)$ $\Rightarrow hasTrajectory(CONST\_1,\ b)$	$\langle LexingtonAve\_4e5, User\_293\rangle$ $\langle NewMingChineseRest, User\_315\rangle$	0.0263 0.0256	1.00 1.00	1.00 1.00
Rule 4.16	$hasCheckin(e,\ a),\ hasCheckin(e,\ b)$ $\Rightarrow withinRadius(a,\ b)$	$\langle\ \rangle$	1.0000	0.47	0.53
Rule 4.17	$hasCheckin(i,\ a),\ hasCheckin(i,\ b),$ $hasTime(a,\ CONST\_0) \Rightarrow withinRadius(a,\ b)$	$\langle Night\rangle$	0.3683	0.48	0.54
Rule 4.18	$hasCheckin(e,\ a),\ hasCheckin(e,\ b),$ $hasWeekday(e,\ CONST\_0) \Rightarrow withinRadius(a,\ b)$	$\langle Weekend\rangle$	0.2502	0.51	0.57
Rule 4.19	$hasCheckin(e,\ a),\ hasCheckin(e,\ b),$ $hasTrajectory(CONST\_0,\ e) \Rightarrow withinRadius(a,\ b)$	$\langle User\_315\rangle$ $\langle User\_185\rangle$	0.0936 0.1084	0.93 0.92	0.93 0.94
Rule 4.20	$hasVenue(c,\ a),\ hasVenue(h,\ a),$ $withinTimeWindow(c,\ h) \Rightarrow hasVenueCategory(a,\ CONST\_0)$	$\langle TravelTransport\rangle$	0.0131	0.27	0.27

Chapter 5
CONCLUSION

We have discussed the opportunities of applying logical rule mining algorithms on ontology-based semantic trajectory data. More specifically, we borrowed from the Knowledge Base Refinement field the state-of-the-art AMIE 3 algorithm and explored how we can apply it to mine rules from the STEP ontology using an intermediate application-specific representation.

This dissertation's initial development was especially inspired by Lecue's work on the STAR-CITY project (LÉCUÉ et al., 2014a). Indeed, there are many opportunities related to mining associations between time, spatial and semantic relations interlinking trajectory events. Mined rules can use such relations to capture interesting mobility patterns and users' preferences. Complementary to Lecue's work, we propose to use Knowledge Refinement algorithms and techniques to leverage the data mining process in ontology-based data.

From the Semantic Trajectory point of view, we use Semantic Web technologies to represent and enrich Location-based Social Network's data. The ontology-based representation is especially used to model relations between different concepts in the trajectory data.

From the Data Mining perspective, we propose to use as a general pattern mining tool an algorithm developed in the context of Knowledge Base Refinement. This algorithm is specifically designed to work on huge volumes of data and mine complex patterns using the ontology facts.

Finally, from a general sense, we propose that many application domains may benefit from using our approach to pattern mining. One important consideration is that this approach makes sense only if we want to explicitly model the relations between the different concepts in a given domain. If there is no such requirement, other approaches such as neural networks could be used. Nonetheless, as we will discuss next, neural networks' representation learning can also be combined with an explicit representation of relations.

The proposed data pipeline mined some interesting patterns in an experiment using two Foursquare datasets. Nonetheless, based on the large number of rules which state facts that are too general, we argue in favor of the design and development of a domain-specific mining algorithm. This algorithm should take advantage of the vast advancements and strategies proposed by the KB refinement community while also considering specific trajectory domain knowledge.

5.1 Open opportunities

Next, we discuss future opportunities based on the acquired experience and experiments, and then highlight and summarize this dissertation's contributions. Some of them have been briefly discussed in Petri and Silva (2020).

Custom relations: We consider essential the ability to add custom, user-defined relations to the data representation. As previously discussed, there are many opportunities in using domain knowledge to improve the contextual representation of a trajectory. This approach could use the ontology-based representation as a framework to inject arbitrarily semantic aspects and inter-relations between complex entities. Techniques similar to MRAR+ (de Oliveira et al., 2019) and other mining and pruning strategies discussed in the next topics could be used to improve the mining performance.

Domain knowledge: We consider that the existence of basic concepts such as a *Moving Object*, *Trajectory*, *Episode*, and *Geographical Entities* (such as a venue) can be explored when mining rules. Indeed, we are mainly interested in how these concepts are related to semantic annotations and themselves. Analyzing the main concepts and semantics should also allow an improved search-space mining strategy, reducing the number of too general rules and allowing mining bigger rules.

Semantic representation: STEP (NOGUEIRA et al., 2018) heavily depends on extending classes, as well as MASTER (MELLO et al., 2019), which depends on adding new semantic instances. We argue that these domain-agnostic representations are not well suited for Horn rules mining since logical rules make sense only if we have meaningful relations between concepts. Therefore, a custom algorithm should *explicitly* or *implicitly* consider a modified data representation strategy. Memory and runtime requirements for converting between representations should be taken into consideration. Another aspect to be analyzed is how much the designed algorithm is coupled with the data representation.

Template guided mining: AMIE explores a top-down rule mining strategy by iteratively applying refinement operators (SUCHANEK et al., 2019). This implies benefits such as applying smart prune strategies. Nonetheless, we consider that alternative approaches that firstly build rule templates could benefit from a user-in-the-loop methodology, where the user validates which templates are interesting to be mined (GHOSH; GHOSH, 2018). This setup could use formally represented domain-knowledge or user interactions to improve the mining search space, yielding better performance while also focusing on the most interesting patterns. The ontology schema could be used to generate candidate rules templates automatically, such as in Wang and Li (2015).

Schema information and taxonomy support: AMIE has very limited support for using the ontology structure information. Some algorithms developed after AMIE tries to deal precisely with such limitation (BARATI et al., 2016). One interesting usage for schema information is to mine rules with multiple granularities. For example, we could define different abstraction levels for the time of the day or venue's categories by combining *rdf:type* and *rdfs:subClassOf*. Also, special care must be taken with special types of relations, such as those symmetric-relations explored in this work.

Numeric and ordinal values: Another interesting feature is being able to mine numerical values. For example, instead of a user-defined threshold for the relation *withinRadius*, a numeric rule could predict the distance threshold that maximizes the rule's confidence metric. Unfortunately, as far as we know, no implementation of a generic system such as Galárraga and Suchanek (2014) has been implemented.

Metrics: PCA (GALÁRRAGA et al., 2013) and other approaches to generate rule's counter-examples (SUCHANEK et al., 2019) should be carefully explored in the context of semantic trajectories. It is clear that missing data will be present in this domain's context, and specific completeness-aware metrics should be developed to take into account the biases and characteristics from the Semantic Trajectory domain.

As investigated in works such as Huang et al. (2004), Monreale et al. (2009), Ghosh and Ghosh (2018) and Zhang et al. (2019), interestingness measures could and should take into account spatial-, temporal- and semantic-aspects. Ideas discussed in subsection 3.5.5 could also be used to better assess a rule's metric given the trajectory domain.

Post-processing and visualization: We have proposed an approach to group similar groups based on metarules. Grouping Horn rules in more complex ways would allow users to better interact with the mined rules. For example, the metarule definition proposed could be relaxed to use subsumption and induce a lattice of related rules. This could then be used in post-processing and visualization steps (RIZK; ELRAGAL, 2012).

Sequential patterns: A common and related task to association rule mining is sequential patterns mining. AMIE and Horn rules are not meant to model sequential facts. However, specific notations and mining approaches could be developed in a similar way to the numerical rules proposed by Galárraga and Suchanek (2014).

Mining other domains: We consider that mining Horn rules using KB refinement algorithms could be extended to mine different data domains using a generic framework. Once semantic and relational data are represented as ontologies, this framework could explore different domain-aware strategies to find complex data patterns using Horn rules' expressiveness.

Embedding-based rule mining: In subsection 2.4.4, we have briefly discussed neural networks and embeddings applied to rule mining. Neural networks have also been applied in trajectory data. Liu et al. (2016) propose a Spatial-Temporal Recurrent Neural Network, based on the sequence modeler Recurrent Neural Network architecture. Time- and space-specific transition matrices are used to compensate for the irregularity in time sampling rate and geographical distance between locations. The neural network model is used to predict users' next location given their location histories, showing state-of-the-art prediction performance in two well-known datasets.

Feng et al. (2017) use learned latent representations to predict the potential users who will visit a given POI in the near future. Using LBSN data and inspired by *word2vec*, the proposed *POI2Vec* uses an embedding network to learn representations of users' sequential check-ins and users' preferences. They show that the learned latent representations can predict the next POI given a user check-ins' history and predict which users may visit a given POI in a given timeframe.

Esuli et al. (2018), also inspired by embedding networks like *word2vec*, propose user-embedding to represent users by their semantic trajectory embeddings. Each segment of a user's trajectory is semantically enriched and then represented as a one-hot encoded vector representation. The aggregation of all segments is used as input to a neural network that learns latent representations for different users with similar behavior. Real data is used to generate a synthetic augmented dataset, where results show the effectiveness of generating latent representations.

We consider that these approaches could be explored and combined with strategies to mine Horn rules using embeddings.

5.2 Contributions

In this section, we list this dissertation's scientific and technical contributions.

5.2.1 Scientific contributions

- Investigate the usage of a KB rule learning algorithm on mining rules from an ontology-based representation of semantic trajectories;
- Investigate the expressiveness power of the mined rules when capturing semantic trajectories patterns;
- Contribute to directing future research on applying KB mining algorithms in ontology-based data;
- Mine association rules from semantic trajectories that capture LBSN users' behaviors;
- Publicly share the technical artifacts used in this work, such as code and software requirements, to allow the reproducibility and extension of this work.

5.2.2 Technical contributions

We mostly use the open-source *trajminer* library and the AMIE 3 implementation (see Section 6.1). Therefore, we make minor and major contributions to these projects.

Trajminer

- Foursquare dataset: we have integrated the NYC and TKY Foursquare datasets (Dingqi Yang et al., 2015) into *trajminer*, which allows easy access and manipulation to the datasets.
- Optimized trajectory processing: we have implemented an alternative version of the data representation layer from *trajminer*, which offer improved data processing performance.

AMIE 3

- Bug report: AMIE presented inconsistent results when mining rules with more than 3 atoms.
- Bug report: AMIE's rule equivalence checker did not deal with all the intended cases.
- Bug fixes and missing code implementation for multiple mining phases.
- Finer controlling on which relations can or should be instantiated.
- Implementation of tests.
- Major performance improvements:
  - Bug fix on the mining operator.
  - Improved usage of the max-length pruning (initially introduced in AMIE+).

5.3 Limitations

In this section, we summarize the limitations of the current approach.

- Using a general-purpose domain-agnostic rule learning system such as AMIE shows to mine a large set of uninteresting rules.
- Without proper mining bias, adding relations that connect a large number of entities makes it harder to distinguish between interesting and uninteresting rules.
- Furthermore, the domain-unawareness greatly impacts the running performance, since too much time is spent processing uninteresting rules.
- We require an explicit application-specific ontology representation to mine interesting rules. Although we propose an automatic process to build it, we note that this process is tightly coupled with the initial data representation chosen.
- AMIE has low support for ontology schema. Therefore, we are not able to mine concepts using features such as a venue's category taxonomy.

5.4 Hypothesis review

As discussed in Chapter 1, this dissertation is based on the investigation of KB rule mining algorithms applied to data not usually mined by them. In particular, we focused our work on mining Semantic Trajectory data.

We have discussed that representations previously proposed in the Semantic Trajectory community are not suitable to be directly mined by our approach. We also explored different issues that arise when using a domain-agnostic mining algorithm. In summary, we have shown that although KB rule mining algorithms can be applied to Semantic Trajectory data, this process tends to require many customizations and modifications.

Therefore, we conclude that the Semantic Trajectory and Knowledge Base Refinement communities have built in recent years a large number of representation and mining approaches that could be put together to mine rules with rich semantic expressiveness from semantic data.

Chapter 6
APPENDICES

6.1 Tech stack

As discussed on Chapter 4, all code and experiments' details are available on GitHub. Multiple code libraries were used to acquire and manipulate data. Here, we list and cite the main technical tools used.

- *AMIE 3* (LAJUS et al., 2020): available on GitHub[1]. The upstream code has many enhancements, such as bug fixes and performance improvements, many of which were contributions of this dissertation. Since we use code outside the scope of AMIE, such as supporting metarules, we maintain a fork on GitHub[2].
- *Owlready2*[3] (LAMY, 2017): used to programmatically manipulate ontologies using a Python API.
- *Trajminer* (PETRY et al., 2019b): trajectory data manipulation tool available on GitHub[4], developed by authors of (PETRY et al., 2019a). We use a performance-improved version, available at GitHub[5].
- *trajectory-data* (PETRY, 2019): scripts to acquire data from the Foursquare API. Available at GitHub[6].
- *pandas*[7] (MCKINNEY, 2010), *scikit-learn*[8] (PEDREGOSA et al., 2011), *matplotlib*[9] (HUNTER, 2007), *seaborn*[10] (WASKOM; team, 2020): tools for manipulating, processing and visualizing tabular data in Python.

[1] <https://github.com/lajus/amie>
[2] <https://github.com/falcaopetri/amie>
[3] <https://owlready2.readthedocs.io/>
[4] <https://github.com/trajminer/trajminer>
[5] <https://github.com/falcaopetri/trajminer>
[6] <https://github.com/lucaspetry/trajectory-data/>
[7] <http://pandas.pydata.org/>
[8] <https://scikit-learn.org/>
[9] <https://matplotlib.org/>
[10] <https://seaborn.pydata.org/>

- *docker*[11] (MERKEL, 2014): tool for deliverying an easy environment for experimental execution and reproducibility.

6.2 Experiment metrics for other datasets

In this section, we complement the data presented in Section 4.2 by showing the KB summaries of the datasets Global Social Foursquare (Table 6.1), Brightkite (Table 6.2), and Gowalla (Table 6.3).

We can observe on them the same analysis explored in Section 4.2: our proposed application-specific representation contains fewer triples and greater semantics when compared to the STEP representation.

[11] <https://www.docker.com/>

Table 6.1 – KB summary for Global Social Foursquare representations.

(a) KB summary for STEP representation.

Relation	Triples	Func.	Inv. Func.	# subjects	# objects
<*hasEpisode*>	15,513,996	0.24	0.09	3,784,266	1,404,976
<*hasFeature*>	5,992,869	0.53	1.00	3,193,478	5,992,869
<*relatesTo*>	4,189,319	0.34	0.72	1,404,976	3,017,679
<*hasSemanticDescription*>	2,208,603	1.00	0.00	2,208,603	34
<*hasTrajectory*>	197,246	0.04	1.00	7,596	197,246
<*hasFriend*>	10,750	0.31	0.31	3,384	3,384
TOTAL	28,112,783	-	-	10,598,919	10,616,188

(b) KB summary for application-specific representation. Values within parentheses are the ones affected by our incompleteness sampling strategy to generate the KB used during the mining phase.

Relation	Triples	Func.	Inv. Func.	# subjects	# objects
<*before*>	5,219,438 (2,549,492)	0.23 (0.31)	0.23 (0.31)	1,207,730 (786,375)	1,207,730 (786,375)
<*hasCheckin*>	1,404,976	0.14	1.00	197,246	1,404,976
<*hasDayOfTheWeek*>	197,246	1.00	0.00	197,246	7
<*hasFriend*>	10,750 (7,524)	0.31 (0.37)	0.31 (0.37)	3,384 (2,793)	3,384 (2,793)
<*hasMonth*>	197,246	1.00	0.00	197,246	12
<*hasVenue*>	1,404,976 (1,045,472)	1.00	0.17 (0.20)	1,404,976 (1,045,472)	233,336 (206,997)
<*hasVenueCategory*>	211,889 (148,322)	1.00	0.00	211,889 (148,322)	10
<*hasTime*>	1,404,976 (983,483)	1.00	0.00	1,404,976 (983,483)	3
<*hasTrajectory*>	197,246	0.04	1.00	7,596	197,246
<*hasWeekday*>	197,246	1.00	0.00	197,246	2
<*withinRadius*>	4,863,650	0.24	0.24	1,165,622	1,165,622
<*withinTimeWindow*>	4,025,932 (1,967,450)	0.30 (0.39)	0.30 (0.39)	1,213,668 (763,314)	1,213,668 (763,314)
TOTAL	19,335,571 (13,759,353)	-	-	1,821,707 (1,740,182)	1,838,976 (1,812,046)

Table 6.2 – KB summary for Brightkite representations.

(a) KB summary for STEP representation.

Relation	Triples	Func.	Inv. Func.	# subjects	# objects
<hasEpisode>	5,319,870	0.26	0.14	1,402,359	721,901
<hasFeature>	2,421,320	0.62	1.00	1,508,114	2,421,320
<relatesTo>	2,130,995	0.34	0.68	721,901	1,442,814
<hasSemanticDescription>	1,018,961	1.00	0.00	1,018,961	24
<hasTrajectory>	99,020	0.02	1.00	2,121	99,020
<hasFriend>	25,178	0.07	0.07	1,767	1,767
TOTAL	11,015,344	-	-	4,653,456	4,686,846

(b) KB summary for application-specific representation. Values within parentheses are the ones affected by our incompleteness sampling strategy to generate the KB used during the mining phase.

Relation	Triples	Func.	Inv. Func.	# subjects	# objects
<before>	2,905,995 (1,424,614)	0.21 (0.29)	0.21 (0.29)	622,881 (406,419)	622,881 (406,419)
<hasCheckin>	721,901	0.14	1.00	99,020	721,901
<hasDayOfTheWeek>	99,020	1.00	0.00	99,020	7
<hasFriend>	25,178 (17,624)	0.07 (0.09)	0.07 (0.09)	1,767 (1,639)	1,767 (1,639)
<hasMonth>	99,020	1.00	0.00	99,020	12
<hasVenue>	721,901 (505,330)	1.00	0.05 (0.06)	721,901 (505,330)	33,720 (31,384)
<hasTime>	721,901 (505,330)	1.00	0.00	721,901 (505,330)	3
<hasTrajectory>	99,020	0.02	1.00	2,121	99,020
<hasWeekday>	99,020	1.00	0.00	99,020	2
<withinRadius>	321,710	0.35	0.35	112,043	112,043
<withinTimeWindow>	1,370,264 (671,218)	0.41 (0.51)	0.41 (0.51)	568,415 (342,575)	568,415 (342,575)
TOTAL	7,184,930 (4,563,807)	-	-	823,042 (768,288)	856,432 (853,968)

Table 6.3 – KB summary for Gowalla representations.

(a) KB summary for STEP representation.

Relation	**Triples**	**Func.**	**Inv. Func.**	**# subjects**	**# objects**
<hasEpisode>	33,204,491	0.08	0.03	2,523,238	907,681
<hasFeature>	3,696,821	0.51	1.00	1,891,376	3,696,821
<relatesTo>	2,710,423	0.33	0.74	907,681	1,995,002
<hasSemanticDescription>	1,173,583	1.00	0.00	1,173,583	24
<hasTrajectory>	88,634	0.04	1.00	3,541	88,634
<hasFriend>	17,182	0.15	0.15	2,500	2,500
TOTAL	40,891,134	-	-	6,499,419	6,690,662

(b) KB summary for application-specific representation. Values within parentheses are the ones affected by our incompleteness sampling strategy to generate the KB used during the mining phase.

Relation	**Triples**	**Func.**	**Inv. Func.**	**# subjects**	**# objects**
<before>	10,431,788 (5,101,434)	0.08 (0.11)	0.08 (0.11)	819,047 (546,812)	819,047 (546,812)
<hasCheckin>	907,681	0.10	1.00	88,634	907,681
<hasDayOfTheWeek>	88,634	1.00	0.00	88,634	7
<hasFriend>	17,182 (12,028)	0.15 (0.19)	0.15 (0.19)	2,500 (2,229)	2,500 (2,229)
<hasMonth>	88,634	1.00	0.00	88,634	12
<hasVenue>	907,681 (635,376)	1.00	0.21 (0.27)	907,681 (635,376)	192,260 (171,142)
<hasTime>	907,681 (635,376)	1.00	0.00	907,681 (635,376)	3
<hasTrajectory>	88,634	0.04	1.00	3,541	88,634
<hasWeekday>	88,634	1.00	0.00	88,634	2
<withinRadius>	10,752,932	0.07	0.07	786,717	786,717
<withinTimeWindow>	11,112,090 (5,428,978)	0.07 (0.10)	0.07 (0.10)	828,840 (547,535)	828,840 (547,535)
TOTAL	35,391,571 (23,828,341)	-	-	999,856 (988,844)	1,191,099 (1,169,710)

BIBLIOGRAPHY

ABBURU, S. A Survey on Ontology Reasoners and Comparison. *International Journal of Computer Applications*, v. 57, p. 7, 2012. Cited on page 28.

AGGARWAL, C. C.; BHUIYAN, M. A.; HASAN, M. A. Frequent Pattern Mining Algorithms: A Survey. In: AGGARWAL, C. C.; HAN, J. (Ed.). *Frequent Pattern Mining*. Cham: Springer International Publishing, 2014. p. 19–64. ISBN 978-3-319-07820-5 978-3-319-07821-2. Cited 3 times on pages 14, 31 e 32.

AGRAWAL, R.; IMIELIŃSKI, T.; SWAMI, A. Mining association rules between sets of items in large databases. In: *Acm Sigmod Record*. [S.l.]: ACM, 1993. v. 22, p. 207–216. ISBN 0-89791-592-5. Cited 3 times on pages 20, 30 e 44.

AGRAWAL, R.; SRIKANT, R.; ROAD, H.; JOSE, S. Fast Algorithms for Mining Association Rules. p. 13, 1994. Cited 2 times on pages 20 e 31.

AKRAMI, F.; SAEEF, M. S.; ZHANG, Q.; HU, W.; LI, C. Realistic Re-evaluation of Knowledge Graph Completion Methods: An Experimental Study. *arXiv:2003.08001 [cs]*, Mar. 2020. Cited on page 78.

ALBANNA, B. H.; MOAWAD, I. F.; MOUSSA, S. M.; SAKR, M. A. Semantic Trajectories: A Survey from Modeling to Application. In: POPOVICH, V.; CLARAMUNT, C.; SCHRENK, M.; KOROLENKO, K.; GENSEL, J. (Ed.). *Information Fusion and Geographic Information Systems (IF&GIS' 2015)*. Cham: Springer International Publishing, 2015. p. 59–76. ISBN 978-3-319-16666-7 978-3-319-16667-4. Cited 6 times on pages 18, 34, 35, 36, 37 e 61.

ATLURI, G.; KARPATNE, A.; KUMAR, V. Spatio-Temporal Data Mining: A Survey of Problems and Methods. *ACM Computing Surveys*, v. 51, n. 4, p. 1–41, Aug. 2018. ISSN 03600300. Cited 3 times on pages 18, 33 e 34.

AUER, S.; BIZER, C.; KOBILAROV, G.; LEHMANN, J.; CYGANIAK, R.; IVES, Z. DBpedia: A Nucleus for a Web of Open Data. In: ABERER, K.; CHOI, K.-S.; NOY, N.; ALLEMANG, D.; LEE, K.-I.; NIXON, L.; GOLBECK, J.; MIKA, P.; MAYNARD, D.; MIZOGUCHI, R.; SCHREIBER, G.; Cudré-Mauroux, P. (Ed.). *The Semantic Web*. [S.l.]: Springer Berlin Heidelberg, 2007. p. 722–735. ISBN 978-3-540-76298-0. Cited 2 times on pages 19 e 29.

BAGLIONI, M.; Fernandes de Macêdo, J. A.; RENSO, C.; TRASARTI, R.; WACHOWICZ, M. Towards semantic interpretation of movement behavior. In: SESTER, M.; BERNARD, L.; PAELKE, V. (Ed.). *Advances in GIScience*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 271–288. ISBN 978-3-642-00318-9. Cited 5 times on pages 37, 39, 40, 41 e 44.

BARATI, M.; BAI, Q.; LIU, Q. SWARM: An Approach for Mining Semantic Association Rules from Semantic Web Data. In: BOOTH, R.; ZHANG, M.-L. (Ed.). *PRICAI 2016: Trends in*

Artificial Intelligence. Cham: Springer International Publishing, 2016. v. 9810, p. 30–43. ISBN 978-3-319-42910-6 978-3-319-42911-3. Cited 3 times on pages 57, 59 e 112.

BARBIERI, D. F.; BRAGA, D.; CERI, S.; VALLE, E. D.; GROSSNIKLAUS, M. C-sparql: A continuous query language for rdf data streams. *International Journal of Semantic Computing*, World Scientific Publishing Co., v. 04, n. 01, p. 3–25, Mar. 2010. ISSN 1793-351X. Cited on page 28.

Berners-Lee, T.; HENDLER, J.; LASSILA, O. et al. The semantic web. *Scientific american*, v. 284, n. 5, p. 28–37, 2001. Cited 2 times on pages 19 e 28.

BIZER, C.; HEATH, T.; Berners-Lee, T. Linked Data - The Story So Far. p. 26, 2009. Cited 2 times on pages 19 e 29.

BOGORNY, V.; AVANCINI, H.; de Paula, B. C.; KUPLICH, C. R.; ALVARES, L. O. Weka-STPM: A Software Architecture and Prototype for Semantic Trajectory Data Mining and Visualization: Weka-STPM: A Software Architecture and Prototype. *Transactions in GIS*, v. 15, n. 2, p. 227–248, Apr. 2011. ISSN 13611682. Cited on page 34.

BOGORNY, V.; HEUSER, C. A.; ALVARES, L. O. A Conceptual Data Model for Trajectory Data Mining. In: HUTCHISON, D.; KANADE, T.; KITTLER, J.; KLEINBERG, J. M.; MATTERN, F.; MITCHELL, J. C.; NAOR, M.; NIERSTRASZ, O.; RANGAN, C. P.; STEFFEN, B.; SUDAN, M.; TERZOPOULOS, D.; TYGAR, D.; VARDI, M. Y.; WEIKUM, G.; FABRIKANT, S. I.; REICHENBACHER, T.; van Kreveld, M.; SCHLIEDER, C. (Ed.). *Geographic Information Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. v. 6292, p. 1–15. ISBN 978-3-642-15299-3 978-3-642-15300-6. Cited on page 37.

BOGORNY, V.; KUIJPERS, B.; ALVARES, L. O. ST-DMQL: A Semantic Trajectory Data Mining Query Language. *International Journal of Geographical Information Science*, v. 23, n. 10, p. 1245–1276, Oct. 2009. ISSN 1365-8816, 1362-3087. Cited 8 times on pages 20, 25, 34, 39, 40, 42, 44 e 84.

BOGORNY, V.; RENSO, C.; de Aquino, A. R.; de Lucca Siqueira, F.; ALVARES, L. O. CONSTAnT - A Conceptual Data Model for Semantic Trajectories of Moving Objects. *Transactions in GIS*, v. 18, n. 1, p. 66–88, Feb. 2014. ISSN 13611682. Cited 3 times on pages 18, 37 e 39.

BOLLACKER, K.; EVANS, C.; PARITOSH, P.; STURGE, T.; TAYLOR, J. Freebase: A collaboratively created graph database for structuring human knowledge. In: *In SIGMOD Conference*. [S.l.: s.n.], 2008. p. 1247–1250. Cited on page 29.

CAMOSSI, E.; VILLA, P.; MAZZOLA, L. Semantic-based Anomalous Pattern Discovery in Moving Object Trajectories. *arXiv:1305.1946 [cs]*, May 2013. Cited 5 times on pages 34, 40, 41, 44 e 46.

CHEN, Y.; WANG, D. Z.; GOLDBERG, S. ScaLeKB: Scalable learning and inference over large knowledge bases. *The VLDB Journal*, v. 25, n. 6, p. 893–918, Dec. 2016. ISSN 0949-877X. Cited on page 49.

CHO, E.; MYERS, S. A.; LESKOVEC, J. Friendship and mobility: User movement in location-based social networks. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '11*. San Diego, California, USA: ACM Press, 2011. p. 1082. ISBN 978-1-4503-0813-7. Cited 3 times on pages 23, 46 e 85.

CHUANGLU, Z. Research on the Semantic Web Reasoning Technology. *AASRI Procedia*, v. 1, p. 87–91, 2012. ISSN 22126716. Cited on page 28.

de Oliveira, F. A.; COSTA, R. L.; GOLDSCHMIDT, R. R.; CAVALCANTI, M. C. Multirelation Association Rule Mining on Datasets of the Web of Data. In: *Proceedings of the XV Brazilian Symposium on Information Systems*. New York, NY, USA: ACM, 2019. (SBSI'19), p. 61:1–61:8. ISBN 978-1-4503-7237-4. Cited 3 times on pages 56, 59 e 112.

DEHASPE, L.; RAEDT, L. D. Mining Association Rules in Multiple Relations. In: *In Proceedings of the 7th International Workshop on Inductive Logic Programming*. [S.l.]: Springer-Verlag, 1997. p. 125–132. Cited 3 times on pages 47, 49 e 59.

Dingqi Yang; Daqing Zhang; ZHENG, V. W.; Zhiyong Yu. Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, v. 45, n. 1, p. 129–142, Jan. 2015. ISSN 2168-2216, 2168-2232. Cited 4 times on pages 23, 46, 85 e 115.

DJENOURI, Y.; DRIAS, H.; HABBAS, Z.; CHEMCHEM, A. Organizing association rules with meta-rules using knowledge clustering. In: *2013 11th International Symposium on Programming and Systems (ISPS)*. [S.l.: s.n.], 2013. p. 109–115. Cited 2 times on pages 23 e 82.

DOU, D.; WANG, H.; LIU, H. Semantic data mining: A survey of ontology-based approaches. In: *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*. Anaheim, CA, USA: IEEE, 2015. p. 244–251. ISBN 978-1-4799-7935-6. Cited 3 times on pages 31, 32 e 33.

du Plessis, L.; ŠKUNCA, N.; DESSIMOZ, C. The what, where, how and why of gene ontology—a primer for bioinformaticians. *Briefings in Bioinformatics*, v. 12, n. 6, p. 723–735, Nov. 2011. ISSN 1467-5463. Cited on page 27.

DŽEROSKI, S. Relational Data Mining. In: MAIMON, O.; ROKACH, L. (Ed.). *Data Mining and Knowledge Discovery Handbook*. Boston, MA: Springer US, 2009. p. 887–911. ISBN 978-0-387-09822-7 978-0-387-09823-4. Cited 4 times on pages 20, 47, 48 e 49.

EBISU, T.; ICHISE, R. Graph Pattern Entity Ranking Model for Knowledge Graph Completion. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 988–997. Cited on page 76.

EHRLINGER, L.; WÖSS, W. Towards a Definition of Knowledge Graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, v. 48, p. 5, 2016. Cited 2 times on pages 28 e 29.

ESULI, A.; PETRY, L. M.; RENSO, C.; BOGORNY, V. Traj2User: Exploiting embeddings for computing similarity of users mobile behavior. *arXiv:1808.00554 [cs]*, Jul. 2018. Cited on page 114.

FAYYAD, U.; Piatetsky-Shapiro, G.; SMYTH, P. From data mining to knowledge discovery in databases. *AI magazine*, v. 17, n. 3, p. 37–37, 1996. ISSN 2371-9621. Cited 2 times on pages 18 e 61.

FENG, S.; CONG, G.; AN, B.; CHEE, Y. M. POI2Vec: Geographical Latent Representation for Predicting Future Visitors. p. 7, 2017. Cited 2 times on pages 83 e 114.

FENG, Z.; ZHU, Y. A Survey on Trajectory Data Mining: Techniques and Applications. *IEEE Access*, v. 4, p. 2056–2067, 2016. ISSN 2169-3536. Cited 7 times on pages 10, 33, 34, 36, 38, 39 e 61.

FERRERO, C. A.; PETRY, L. M.; ALVARES, L. O.; da Silva, C. L.; ZALEWSKI, W.; BOGORNY, V. MasterMovelets: Discovering heterogeneous movelets for multiple aspect trajectory classification. *Data Mining and Knowledge Discovery*, v. 34, n. 3, p. 652–680, May 2020. ISSN 1384-5810, 1573-756X. Cited 4 times on pages 10, 17, 18 e 87.

FILETO, R.; BOGORNY, V.; MAY, C.; KLEIN, D. Semantic Enrichment and Analysis of Movement Data: Probably it is just Starting! p. 8, 2015. Cited on page 45.

FILETO, R.; MAY, C.; RENSO, C.; PELEKIS, N.; KLEIN, D.; THEODORIDIS, Y. The Baquara2 knowledge-based framework for semantic enrichment and analysis of movement data. *Data & Knowledge Engineering*, v. 98, p. 104–122, Jul. 2015. ISSN 0169023X. Cited 7 times on pages 19, 20, 23, 35, 37, 45 e 64.

Fournier-Viger, P.; LIN, J. C.-W. A Survey of Sequential Pattern Mining. p. 24, 2017. Cited on page 33.

Fournier-Viger, P.; LIN, J. C.-W.; VO, B.; CHI, T. T.; ZHANG, J.; LE, H. B. A survey of itemset mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, v. 7, n. 4, p. e1207, Jul. 2017. ISSN 19424787. Cited 4 times on pages 30, 32, 33 e 42.

GALÁRRAGA, L.; SUCHANEK, F. M. Towards a numerical rule mining language. In: . [S.l.: s.n.], 2014. Cited 3 times on pages 22, 53 e 113.

GALÁRRAGA, L.; TEFLIOUDI, C.; HOSE, K.; SUCHANEK, F. M. Fast rule mining in ontological knowledge bases with AMIE+. *The VLDB Journal*, v. 24, n. 6, p. 707–730, Dec. 2015. ISSN 1066-8888, 0949-877X. Cited on page 49.

GALÁRRAGA, L. A.; TEFLIOUDI, C.; HOSE, K.; SUCHANEK, F. AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In: *Proceedings of the 22nd International Conference on World Wide Web - WWW '13*. Rio de Janeiro, Brazil: ACM Press, 2013. p. 413–422. ISBN 978-1-4503-2035-1. Cited 13 times on pages 12, 14, 19, 28, 29, 47, 49, 50, 51, 52, 53, 81 e 113.

GANGEMI, A. Ontology design patterns for semantic web content. In: *International Semantic Web Conference*. [S.l.]: Springer, 2005. p. 262–276. Cited on page 64.

GENG, L.; HAMILTON, H. J. Interestingness measures for data mining: A survey. *ACM Computing Surveys*, v. 38, n. 3, p. 9–es, Sep. 2006. ISSN 03600300. Cited on page 31.

GHOSH, S.; GHOSH, S. K. Exploring human movement behaviour based on mobility association rule mining of trajectory traces. In: Springer. *International Conference on Intelligent Systems Design and Applications*. [S.l.], 2017. p. 451–463. Cited on page 81.

GHOSH, S.; GHOSH, S. K. Exploring Human Movement Behaviour Based on Mobility Association Rule Mining of Trajectory Traces. In: ABRAHAM, A.; MUHURI, P. K.; MUDA, A. K.; GANDHI, N. (Ed.). *Intelligent Systems Design and Applications*. Cham: Springer International Publishing, 2018. v. 736, p. 451–463. ISBN 978-3-319-76347-7 978-3-319-76348-4. Cited 3 times on pages 42, 112 e 113.

GIANNOTTI, F.; NANNI, M.; PINELLI, F.; PEDRESCHI, D. Trajectory pattern mining. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [S.l.: s.n.], 2007. p. 330–339. Cited on page 40.

GIL, R.; NABO, B.; FILETO, R.; NANNI, M.; RENSO, C. Annotating Trajectories by Fusing them with Social Media Users' Posts. 2014. Cited 2 times on pages 45 e 46.

GOETHALS, B.; Van den Bussche, J. Relational association rules: Getting Warmer. In: *Pattern Detection and Discovery*. [S.l.]: Springer, 2002. p. 125–139. Cited 2 times on pages 47 e 49.

GRUBER, T. R. A translation approach to portable ontology specifications. *Knowledge Acquisition*, v. 5, n. 2, p. 199–220, Jun. 1993. ISSN 1042-8143. Cited on page 27.

HU, Y.; JANOWICZ, K.; CARRAL, D.; SCHEIDER, S.; KUHN, W.; Berg-Cross, G.; HITZLER, P.; DEAN, M.; KOLAS, D. A Geo-ontology Design Pattern for Semantic Trajectories. In: HUTCHISON, D.; KANADE, T.; KITTLER, J.; KLEINBERG, J. M.; MATTERN, F.; MITCHELL, J. C.; NAOR, M.; NIERSTRASZ, O.; RANGAN, C. P.; STEFFEN, B.; SUDAN, M.; TERZOPOULOS, D.; TYGAR, D.; VARDI, M. Y.; WEIKUM, G.; TENBRINK, T.; STELL, J.; GALTON, A.; WOOD, Z. (Ed.). *Spatial Information Theory*. Cham: Springer International Publishing, 2013. v. 8116, p. 438–456. ISBN 978-3-319-01789-1 978-3-319-01790-7. Cited on page 64.

HUANG, Y.; SHEKHAR, S.; XIONG, H. Discovering colocation patterns from spatial data sets: A general approach. *IEEE Transactions on Knowledge and Data Engineering*, v. 16, n. 12, p. 1472–1485, Dec. 2004. ISSN 1041-4347. Cited 4 times on pages 39, 40, 44 e 113.

HUNTER, J. D. Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering*, v. 9, n. 3, p. 90–95, May 2007. ISSN 1558-366X. Cited on page 117.

KAMBER, M.; HAN, J.; CHIANG, J. Metarule-guided mining of multi-dimensional association rules using data cubes. In: *KDD*. [S.l.: s.n.], 1997. v. 97, p. 207. Cited 2 times on pages 23 e 82.

KHOSHAHVAL, S.; FARNAGHI, M.; TALEAI, M. SPATIO-TEMPORAL PATTERN MINING ON TRAJECTORY DATA USING ARM. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W4, p. 395–399, Sep. 2017. ISSN 2194-9034. Cited 3 times on pages 20, 40 e 84.

LAJUS, J.; GALÁRRAGA, L.; SUCHANEK, F. Fast and exact rule mining with AMIE 3. In: *European Semantic Web Conference*. [S.l.: s.n.], 2020. p. 36–52. Cited 8 times on pages 19, 23, 47, 49, 50, 59, 76 e 117.

LAMY, J.-B. Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. *Artificial Intelligence in Medicine*, v. 80, p. 11–28, Jul. 2017. ISSN 09333657. Cited 2 times on pages 28 e 117.

LANDSEA, C. W.; FRANKLIN, J. L. Atlantic Hurricane Database Uncertainty and Presentation of a New Database Format. *Monthly Weather Review*, v. 141, n. 10, p. 3576–3592, Jan. 2013. ISSN 0027-0644. Cited on page 46.

LAUBE, P. *Computational Movement Analysis*. New York: Springer, 2014. ISBN 978-3-319-10267-2. Cited on page 17.

LAUBE, P. The low hanging fruit is gone: Achievements and challenges of computational movement analysis. *SIGSPATIAL Special*, v. 7, n. 1, p. 3–10, 2015. Cited 2 times on pages 35 e 86.

LÉCUÉ, F. Scalable Maintenance of Knowledge Discovery in an Ontology Stream. p. 7, Jun. 2015. Cited on page 54.

LÉCUÉ, F.; PAN, J. Z. Predicting Knowledge in an Ontology Stream. p. 8, 2013. Cited 2 times on pages 54 e 59.

LÉCUÉ, F.; PAN, J. Z. Consistent knowledge discovery from evolving ontologies. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2015. Cited on page 54.

LÉCUÉ, F.; Tallevi-Diotallevi, S.; HAYES, J.; TUCKER, R.; BICER, V.; SBODIO, M.; TOMMASI, P. Smart traffic analytics in the semantic web with STAR-CITY: Scenarios, system and lessons learned in Dublin City. *Journal of Web Semantics*, v. 27-28, p. 26–33, Aug. 2014. ISSN 15708268. Cited 3 times on pages 54, 55 e 111.

LÉCUÉ, F.; TUCKER, R.; BICER, V.; TOMMASI, P.; SBODIO, M. Predicting Severity of Road Traffic Congestion using Semantic Web Technologies. p. 15, 2014. Cited on page 54.

LÉCUÉ, F.; TUCKER, R.; Tallevi-Diotallevi, S.; NAIR, R.; GKOUFAS, Y.; LIGUORI, G.; BORIONI, M.; RADEMAKER, A.; BARBOSA, L. Semantic Traffic Diagnosis with STAR-CITY: Architecture and Lessons Learned from Deployment in Dublin, Bologna, Miami and Rio. In: MIKA, P.; TUDORACHE, T.; BERNSTEIN, A.; WELTY, C.; KNOBLOCK, C.; VRANDEČIĆ, D.; GROTH, P.; NOY, N.; JANOWICZ, K.; GOBLE, C. (Ed.). *The Semantic Web – ISWC 2014*. [S.l.]: Springer International Publishing, 2014. p. 292–307. ISBN 978-3-319-11915-1. Cited on page 54.

LÉCUÉ, F.; WU, J. Explaining and predicting abnormal expenses at large scale using knowledge graph based reasoning. *Journal of Web Semantics*, v. 44, p. 89–103, May 2017. ISSN 15708268. Cited on page 55.

LIU, Q.; WU, S.; WANG, L.; TAN, T. Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts. p. 7, 2016. Cited 3 times on pages 34, 83 e 113.

MCKINNEY, W. Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, p. 56–61, 2010. Cited on page 117.

MEILICKE, C.; FINK, M.; WANG, Y.; RUFFINELLI, D.; GEMULLA, R.; STUCKENSCHMIDT, H. Fine-Grained Evaluation of Rule- and Embedding-Based Systems for Knowledge Graph Completion. In: VRANDEČIĆ, D.; BONTCHEVA, K.; Suárez-Figueroa, M. C.; PRESUTTI, V.; CELINO, I.; SABOU, M.; KAFFEE, L.-A.; SIMPERL, E. (Ed.). *The Semantic Web – ISWC 2018*. Cham: Springer International Publishing, 2018. v. 11136, p. 3–20. ISBN 978-3-030-00670-9 978-3-030-00671-6. Cited on page 78.

MELLO, R. d. S.; BOGORNY, V.; ALVARES, L. O.; SANTANA, L. H. Z.; FERRERO, C. A.; FROZZA, A. A.; SCHREINER, G. A.; RENSO, C. MASTER: A multiple aspect view on trajectories. *Transactions in GIS*, v. 23, n. 4, p. 805–822, 2019. ISSN 1467-9671. Cited 5 times on pages 20, 23, 37, 38 e 112.

MERKEL, D. Docker: Lightweight linux containers for consistent development and deployment. *Linux journal*, v. 2014, n. 239, p. 2, 2014. Cited on page 118.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G. S.; DEAN, J. Distributed Representations of Words and Phrases and their Compositionality. In: BURGES, C. J. C.; BOTTOU, L.; WELLING, M.; GHAHRAMANI, Z.; WEINBERGER, K. Q. (Ed.). *Advances in Neural Information Processing Systems 26*. [S.l.]: Curran Associates, Inc., 2013. p. 3111–3119. Cited on page 58.

MITCHELL, T.; COHEN, W.; HRUSCHKA, E.; TALUKDAR, P.; YANG, B.; BETTERIDGE, J.; CARLSON, A.; DALVI, B.; GARDNER, M.; KISIEL, B.; KRISHNAMURTHY, J.; LAO, N.; MAZAITIS, K.; MOHAMED, T.; NAKASHOLE, N.; PLATANIOS, E.; RITTER, A.; SAMADI, M.; SETTLES, B.; WANG, R.; WIJAYA, D.; GUPTA, A.; CHEN, X.; SAPAROV, A.; GREAVES, M.; WELLING, J. Never-ending Learning. *Commun. ACM*, v. 61, n. 5, p. 103–115, Apr. 2018. ISSN 0001-0782. Cited on page 29.

MONREALE, A.; PINELLI, F.; TRASARTI, R.; GIANNOTTI, F. WhereNext: A location predictor on trajectory pattern mining. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '09*. Paris, France: ACM Press, 2009. p. 637. ISBN 978-1-60558-495-9. Cited 4 times on pages 39, 40, 81 e 113.

MOUSAVI, A.; HUNTER, A.; AKBARI, M. Using ontology based semantic association rule mining in location based services. *International Journal of Data Mining & Knowledge Management Process*, v. 6, p. 1–14, 2016. Cited 5 times on pages 20, 40, 42, 44 e 84.

NOGUEIRA, T. P.; BRAGA, R. B.; de Oliveira, C. T.; MARTIN, H. FrameSTEP: A framework for annotating semantic trajectories based on episodes. *Expert Systems with Applications*, v. 92, p. 533–545, Feb. 2018. ISSN 09574174. Cited 10 times on pages 10, 20, 23, 34, 37, 38, 64, 65, 66 e 112.

OMRAN, P. G.; WANG, K.; WANG, Z. Scalable Rule Learning via Learning Representation. p. 7, 2018. Cited 3 times on pages 29, 58 e 59.

ORTONA, S.; MEDURI, V. V.; PAPOTTI, P. Robust Discovery of Positive and Negative Rules in Knowledge Bases. In: *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. Paris: IEEE, 2018. p. 1168–1179. ISBN 978-1-5386-5520-7. Cited on page 49.

PARENT, C.; SPACCAPIETRA, S.; RENSO, C.; ANDRIENKO, G.; ANDRIENKO, N.; BOGORNY, V.; DAMIANI, M. L.; Gkoulalas-Divanis, A.; MACEDO, J.; PELEKIS, N.; THEODORIDIS, Y.; YAN, Z. Semantic trajectories modeling and analysis. *ACM Computing Surveys*, v. 45, n. 4, p. 32, 2013. Cited 3 times on pages 33, 34 e 37.

PAULHEIM, H. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, v. 8, n. 3, p. 489–508, Dec. 2016. ISSN 22104968, 15700844. Cited 4 times on pages 19, 29, 46 e 47.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, É. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, v. 12, n. 85, p. 2825–2830, 2011. ISSN 1533-7928. Cited on page 117.

PELEKIS, N.; SIDERIDIS, S.; TAMPAKIS, P.; THEODORIDIS, Y. Simulating Our LifeSteps by Example. *ACM Transactions on Spatial Algorithms and Systems*, v. 2, n. 3, p. 1–39, Oct. 2016. ISSN 23740353. Cited 2 times on pages 45 e 46.

PERRY, M.; HERRING, J. OGC GeoSPARQL - A Geographic Query Language for RDF Data. p. 75, 2012. Cited on page 28.

PETRI, A. C. F.; SILVA, D. F. Towards logical association rule mining on ontology-based semantic trajectories. In: *ICMLA (in press)*. [S.l.: s.n.], 2020. Cited 4 times on pages 20, 25, 50 e 112.

PETRY, L. M. *Lucaspetry/Trajectory-Data*. 2019. Cited on page 117.

PETRY, L. M.; FERRERO, C. A.; ALVARES, L. O.; RENSO, C.; BOGORNY, V. Towards semantic-aware multiple-aspect trajectory similarity measuring. *Transactions in GIS*, Jun. 2019. ISSN 13611682. Cited 6 times on pages 36, 39, 45, 86, 87 e 117.

PETRY, L. M. et al. *Trajminer*. 2019. Cited 2 times on pages 25 e 117.

RAMEZANI, R.; SARAEE, M.; NEMATBAKHSH, M. MRAR : Mining multi-relation association rules. *Journal of Computing and Security*, v. 1, n. 2, p. 133–158, Sep. 2014. Cited 5 times on pages 30, 48, 55, 56 e 59.

REN, Y.; PAN, J. Z. Optimising ontology stream reasoning with truth maintenance system. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management - CIKM '11*. Glasgow, Scotland, UK: ACM Press, 2011. p. 831. ISBN 978-1-4503-0717-8. Cited on page 54.

RENSO, C.; BAGLIONI, M.; de Macedo, J. A. F.; TRASARTI, R.; WACHOWICZ, M. How you move reveals who you are: Understanding human behavior by analyzing trajectory data. *Knowledge and Information Systems*, v. 37, n. 2, p. 331–362, Nov. 2013. ISSN 0219-3116. Cited 8 times on pages 17, 18, 20, 28, 35, 40, 41 e 44.

RISTOSKI, P.; PAULHEIM, H. Semantic Web in data mining and knowledge discovery: A comprehensive survey. *Journal of Web Semantics*, v. 36, p. 1–22, Jan. 2016. ISSN 15708268. Cited 3 times on pages 20, 29 e 32.

RIZK, A.; ELRAGAL, A. Trajectory Data Analysis in Support of Understanding Movement Patterns: A Data Mining Approach. In: *AMCIS*. [S.l.: s.n.], 2012. Cited 5 times on pages 20, 40, 42, 44 e 113.

SENGSTOCK, C.; GERTZ, M. Spatial Itemset Mining: A Framework to Explore Itemsets in Geographic Space. In: HUTCHISON, D.; KANADE, T.; KITTLER, J.; KLEINBERG, J. M.; MATTERN, F.; MITCHELL, J. C.; NAOR, M.; NIERSTRASZ, O.; RANGAN, C. P.; STEFFEN, B.; SUDAN, M.; TERZOPOULOS, D.; TYGAR, D.; VARDI, M. Y.; WEIKUM, G.; CATANIA, B.; GUERRINI, G.; POKORNÝ, J. (Ed.). *Advances in Databases and Information Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. v. 8133, p. 148–161. ISBN 978-3-642-40682-9 978-3-642-40683-6. Cited 4 times on pages 39, 40, 41 e 44.

SENOZETNIK, M.; BRADESKO, L.; SUBIC, T.; HERGA, Z.; URBANCIC, J.; SKRABA, P.; MLADENIC, D. Estimating point-of-interest rating based on visitors geospatial behaviour. *Computer Science and Information Systems*, v. 16, n. 1, p. 131–154, 2019. ISSN 1820-0214, 2406-1018. Cited on page 45.

SILVA, T. H.; VIANA, A. C.; BENEVENUTO, F.; VILLAS, L.; SALLES, J.; LOUREIRO, A.; QUERCIA, D. Urban Computing Leveraging Location-Based Social Network Data: A Survey. *ACM Computing Surveys*, v. 52, n. 1, p. 17:1–17:39, Feb. 2019. ISSN 0360-0300. Cited 2 times on pages 17 e 33.

SPACCAPIETRA, S.; PARENT, C.; DAMIANI, M. L.; de Macedo, J. A.; PORTO, F.; VANGENOT, C. A conceptual view on trajectories. *Data & Knowledge Engineering*, v. 65, n. 1, p. 126–146, Apr. 2008. ISSN 0169023X. Cited 3 times on pages 34, 36 e 37.

SRIKANT, R.; AGRAWAL, R. Mining Quantitative Association Rules in Large Relational Tables. In: *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: ACM, 1996. (SIGMOD '96), p. 1–12. ISBN 978-0-89791-794-0. Cited on page 30.

SRIKANT, R.; AGRAWAL, R. Mining generalized association rules. *Future Generation Computer Systems*, v. 13, n. 2-3, p. 161–180, Nov. 1997. ISSN 0167739X. Cited 2 times on pages 30 e 33.

STOICA, I.; SONG, D.; POPA, R. A.; PATTERSON, D.; MAHONEY, M. W.; KATZ, R.; JOSEPH, A. D.; JORDAN, M.; HELLERSTEIN, J. M.; GONZALEZ, J. E.; GOLDBERG, K.; GHODSI, A.; CULLER, D.; ABBEEL, P. A Berkeley View of Systems Challenges for AI. *arXiv:1712.05855 [cs]*, Dec. 2017. Cited on page 58.

SUCHANEK, F. M.; ABITEBOUL, S.; SENELLART, P. PARIS: Probabilistic alignment of relations, instances, and schema. *Proceedings of the VLDB Endowment*, v. 5, n. 3, p. 157–168, Nov. 2011. ISSN 2150-8097. Cited on page 50.

SUCHANEK, F. M.; KASNECI, G.; WEIKUM, G. Yago: A core of semantic knowledge. In: *Proceedings of the 16th International Conference on World Wide Web*. New York, NY, USA: ACM, 2007. (WWW '07), p. 697–706. ISBN 978-1-59593-654-7. Cited 2 times on pages 19 e 29.

SUCHANEK, F. M.; LAJUS, J.; BOSCHIN, A.; WEIKUM, G. Knowledge Representation and Rule Mining in Entity-Centric Knowledge Bases. In: KRÖTZSCH, M.; STEPANOVA, D. (Ed.). *Reasoning Web. Explainable Artificial Intelligence*. Cham: Springer International Publishing, 2019. v. 11810, p. 110–152. ISBN 978-3-030-31422-4 978-3-030-31423-1. Cited 9 times on pages 28, 29, 30, 46, 51, 58, 81, 112 e 113.

TAKIMOTO, Y.; SUGIURA, K.; ISHIKAWA, Y. Extraction of Frequent Patterns Based on Users' Interests from Semantic Trajectories with Photographs. In: *Proceedings of the 21st International Database Engineering & Applications Symposium*. New York, NY, USA: ACM, 2017. (IDEAS 2017), p. 219–227. ISBN 978-1-4503-5220-8. Cited 2 times on pages 18 e 46.

TANON, T. P.; STEPANOVA, D.; RAZNIEWSKI, S.; MIRZA, P.; WEIKUM, G. Completeness-aware Rule Learning from Knowledge Graphs. p. 16, 2017. Cited 2 times on pages 52 e 81.

VERHEIN, F.; CHAWLA, S. Mining spatio-temporal patterns in object mobility databases. *Data Mining and Knowledge Discovery*, v. 16, n. 1, p. 5–38, Feb. 2008. ISSN 1384-5810, 1573-756X. Cited 4 times on pages 39, 40, 44 e 81.

VRANDEČIĆ, D.; KRÖTZSCH, M. Wikidata: A free collaborative knowledge base. *Commun. ACM*, v. 57, n. 10, p. 78–85, Sep. 2014. ISSN 0001-0782. Cited 2 times on pages 19 e 29.

W3C. *RDF - Semantic Web Standards*. 2014. Https://www.w3.org/RDF/. Cited on page 28.

WANG, H.; JIN, P.; ZHAO, L.; ZHANG, L.; YUE, L. Generating Semantic-Based Trajectories for Indoor Moving Objects. In: WANG, L.; JIANG, J.; LU, J.; HONG, L.; LIU, B. (Ed.). *Web-Age Information Management*. [S.l.]: Springer Berlin Heidelberg, 2012. p. 13–25. ISBN 978-3-642-28635-3. Cited on page 46.

WANG, Z.; LI, J. RDF2Rules: Learning Rules from RDF Knowledge Bases by Mining Frequent Predicate Cycles. *arXiv:1512.07734 [cs]*, Dec. 2015. Cited on page 112.

WASKOM, M.; team, t. seaborn development. *Mwaskom/Seaborn*. 2020. Zenodo. Cited on page 117.

YAN, Z.; CHAKRABORTY, D.; PARENT, C.; SPACCAPIETRA, S.; ABERER, K. Semantic trajectories: Mobility data computation and annotation. *ACM Transactions on Intelligent Systems and Technology*, v. 4, n. 3, p. 1, Jun. 2013. ISSN 21576904. Cited on page 36.

YANG, D.; QU, B.; YANG, J.; Cudre-Mauroux, P. Revisiting User Mobility and Social Relationships in LBSNs: A Hypergraph Embedding Approach. In: *The World Wide Web Conference on - WWW '19*. San Francisco, CA, USA: ACM Press, 2019. p. 2147–2157. ISBN 978-1-4503-6674-8. Cited 3 times on pages 23, 83 e 85.

YANG, D.; QU, B.; YANG, J.; Cudre-Mauroux, P. LBSN2Vec++: Heterogeneous Hypergraph Embedding for Location-Based Social Networks. *IEEE Transactions on Knowledge and Data Engineering*, p. 1–1, 2020. ISSN 1558-2191. Cited 4 times on pages 23, 34, 83 e 85.

YANG, D.; ZHANG, D.; QU, B.; Cudré-Mauroux, P. PrivCheck: Privacy-preserving check-in data publishing for personalized location based services. In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '16*. Heidelberg, Germany: ACM Press, 2016. p. 545–556. ISBN 978-1-4503-4461-6. Cited on page 86.

YUAN, J.; ZHENG, Y.; ZHANG, C.; XIE, W.; XIE, X.; SUN, G.; HUANG, Y. T-drive: Driving Directions Based on Taxi Trajectories. In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. New York, NY, USA: ACM, 2010. (GIS '10), p. 99–108. ISBN 978-1-4503-0428-3. Cited on page 45.

ZHANG, W.; WANG, X.; HUANG, Z. A System of Mining Semantic Trajectory Patterns from GPS Data of Real Users. *Symmetry*, v. 11, n. 7, p. 889, Jul. 2019. ISSN 2073-8994. Cited 3 times on pages 40, 42 e 113.

ZHENG, Y. Trajectory Data Mining: An Overview. *ACM Transactions on Intelligent Systems and Technology*, v. 6, n. 3, p. 1–41, May 2015. ISSN 21576904. Cited 7 times on pages 33, 34, 36, 38, 39, 45 e 46.

ZHENG, Y.; XIE, X.; MA, W.-Y. GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. *IEEE Data Engineering Bulletin*, v. 33, n. 2, p. 32–39, 2010. Cited on page 45.