This chapter covers the background areas and related work necessary to understand the contributions of this dissertation. It discusses the current state of the effort to incorporate domain knowledge in data mining. In addition, it describes the various formalisms proposed in the literature to use graphs in tackling data mining and knowledge representation problems. We conclude with a high-level summary of these background areas.

0.1. Domain Knowledge in Data Mining

Domain knowledge relates to information about a specific domain or data that is collected from previous systems or documentation, or elicited from domain experts. In the rest of the section, we highlight a body of studies that aims at exploring ways to employ domain knowledge in data mining. The results from these studies strongly attest to the positive influence of domain knowledge. Domain knowledge can affect the discovery process within a data mining system in at least two ways. First, it can make patterns more visible by generalizing the attribute values, and second, it can constrain the search space as well as the rule space.

In order to effectively summarize and compare different previously proposed systems, we propose a reference framework to classify different kinds of domain knowledge at a very high abstraction level as detailed in the following.

- Knowledge about the domain: This category contains information related to a specific domain of discourse, usually obtained from either domain experts or previous data mining processes. Examples of such knowledge include concept hierarchy, integrity constraints, *etc.*

- Knowledge about the data: This category contains information about dataset, including how it is generated, transformed and evolved. This knowledge is obtained from data producers (people who carry out experiments or collect data) or database

managers. *e.g.*, in a database of spatial information, one of the images may have been recorded with a very skew angle on the object. When processing the database the discovery process must take this information into account.

– Knowledge about the data mining process: This category contains information pertaining to specific data mining sessions, including goals, parameters and variables related to the experiment. *e.g.*, attributes of interest within data, roles of attributes as being whether antecedents or consequences in association rule mining, and the measure of interestingness for discovered patterns.

The summarized work can be divided into two groups. The first group does not explicitly leverage any knowledge representation approaches to model domain knowledge. The second group explores mainly ontological knowledge (concept hierarchy) and uses formal ontology languages to encode such knowledge. The kind of domain knowledge involved in the first group is broader which covers all categories discussed in the above reference classification scheme. However, it is achieved at the cost of less formality which often result in ad-hoc expression of domain knowledge that has a very application-specific form, scop and granularity.

In one of the earliest studies on the subject, Pazzani and Kibler [1] developed a general purpose relational learning algorithm called FOCL, which combines explanation-based and inductive learning. In a later study, they conducted an experiment comparing FOCL with a domain theory to FOCL without a domain theory. A partial knowledge base of an expert system was used as the domain theory. They found that incorporating domain theory significantly reduced misclassification costs when larger training sets were used.

In another study, Ambrosino and Buchanan [2] examined the effects of adding domain knowledge to a rule induction program for predicting the risk of mortality in patients with community–acquired pneumonia. They developed a graphical data exploration tool for

domain experts to encode domain knowledge and interact with the data mining process. The domain experts participated in two stages of mining. They were first asked to modify the existing set of attributes according to their domain knowledge, and then they were prompted with mining results and were able to modify the mined models (rules) directly. The experiment contained an experimental where the domain knowledge was incorporated as mentioned above, and a control group without domain knowledge. The experimental group performed significantly better (lower percent mean error) than control group.

Sinha and Zhao [3] conducted an extensive comparative study on the performance of seven data mining classification methods—naive Bayes, logistic regression, decision tree, decision table, neural network k-nearest neighbor, and support vector machine—with and without incorporating domain knowledge. The application they focused on was in the domain of indirect bank lending. An expert system capturing a lending expert's knowledge of rating a borrower's credit is used in combination with data mining to study if the incorporation of domain knowledge improves classification performance. In their study, the domain knowledge used was partial, meaning that it could only lead to intermediate results but was not sufficient to make the final prediction. They cascaded the expert system with a data mining classifier. The experiment adopted an experimental vs. control paradigm, similar to Ambrosino et al.'s early experiment in 1999. The prediction proposed by the expert system was added to other inputs. Classifiers built using input data enhanced by the expert system's output formed the experimental group. For the control group, classifiers were built using the original set of input attributes (bypassing the expert system). Their results showed that incorporation of domain knowledge significantly improves classification results with respect to both misclassification cost and AUC (Area Under Curve). Hence they concluded by calling for more attention in combining domain knowledge and data mining. They articulated that, in knowledge engineering, the focus is on the knowledge of a human expert in a specific problem area. On the other hand, the focus of data mining is on the data available in an organization. Expert

systems and data mining methods could play complementary roles in situations where both knowledge and data are available.

Hirsh and Noordewier [4] argued that if learning is to be successful, the training data must be encoded in a form that lets the learner recognize underlying regularities. The application domain they focused on was the problem DNA sequence classification. They proposed to use background knowledge of molecular biology to re-express data in terms of higher-level features that molecular biologists use when discussing nucleic-acid sequences. The high lever features were Boolean valued, representing the presence or absence of the feature in a given DNA sequence. Using C4.5 decision trees and backprop neural networks, they conducted experiments with and without the higher-level features. For both learning methods, the use of higher-level features resulted in significantly lower error rates.

Pohle [5] contended that data mining techniques are good at generating useful statistics and finding patterns in large volumes of data, but "they are not very smart in interpreting these results, which is crucial for turning them into interesting, understandable and actionable knowledge". The author viewed the lack of sophisticated tool to support incorporating human domain knowledge into the mining process as being the main factor responsible for the limitation. They also pointed out that ontologies were valuable technologies to incorporate domain knowledge and thus they propose to exploit ontologies when integrating knowledge mined from knowledge discovery process to an existing knowledge base.

Kopanas *et al.* [6] conducted large scale data mining experiment exploring the role of domain knowledge in different phases of a large-scale data mining project, using a case study of customer insolvency in the telecommunications industry. They argued against the claim that data mining approaches eventually will automate the process and lead to discovery of knowledge from data with little or no support of domain experts and domain knowledge. For each stage in data mining they identified types of domain knowledge involved to either

improve the performance or, in some case, make data mining process possible at all. They found that though domain knowledge plays a critical role mainly in the initial and final phases of the project, it influences the other phases to some degree as well. For example, in problem definition stage, domain knowledge involves business and domain requirements and other implicit, tacit knowledge. In data preparation stage, the useful domain knowledge involves semantics of corporate database. In data preprocessing and transformation stage, domain knowledge includes tacit and implicit knowledge for inferences. In feature and algorithm selection stage, main type of knowledge involves how to interpret selected features. In mining stage, domain knowledge focuses on inspection of discovered knowledge. In evaluation stage, domain knowledge defines performance criteria related to business objectives. In fielding knowledge base stage (incorporating mined knowledge with existing knowledge base), domain knowledge provides supplementary information for implementing the fusion.

In another study, Weiss *et al.* [7] combined an expert system with a data mining method for generating better sales leads. They developed an expert system that interviews executives of small and medium-sized companies and, based on their responses, recommends promising sales leads. The question-answer pairs and the recommended solutions were stored as examples to be mined by the method of rule induction. The study demonstrated how a knowledge base can be used to guide a machine learning program. The techniques developed in the study would be useful for consultation systems whose questions have different costs of acquisition.

Daniels *et al.* [8] demonstrated that data mining systems can be successfully combined with explicit domain knowledge. They pointed out that in theory there are two extreme situations that may occur with respect to the availability of domain knowledge. The first is that no prior knowledge whatsoever is available. The second is all relationship is known with certainty, up to a limited number of parameters. They then claimed that their study was positioned somewhere between these extremes. The authors focused on a special type

of a priori knowledge, monotonicity, i.e., the sign of relationship between the dependent and independent variables, for economic decision problems. Prior knowledge was implemented as monotonicity constraints in decision tree and neural network classifiers. Addition of the knowledge resulted in smaller decision trees, and smaller variations of error R2 on the training and test sets for neural networks. The authors also claimed that the framework developed might serve as a tool to implement normative requirements. However, since monotonicity constraints were incorporated in the decision tree and neural networks by designing specific algorithms, it is not obvious how to generalize the algorithm design process to include other normative domain knowledge.

Yoon *et al.* [9] studied semantic query optimization, a field that endeavors to optimize data mining queries by taking advantage of domain knowledge. The authors demonstrated that significant cost reduction can be achieved by reformulating a query into a less expensive yet equivalent query that produces the same answer as the original one. They identified that in most cases, exhaustive analysis of data is infeasible. It is often necessary to perform a relatively constrained search on a specific subset of data for desired knowledge. The domain knowledge they utilized was classified into three categories, interfiled, category, and correlation, all of which can be represented in rule forms. When a data mining query is received, they first identify domain knowledge that is relevant to the query, and transform it accordingly. On the other hand, to select relevant domain knowledge without an exhaustive search of all domain knowledge, they developed a method that built tables for domain knowledge indexed by attributes.

Vikram and Nagpal [10] developed an iterative association rule mining algorithm to integrate user's domain knowledge with association rule mining. The knowledge they request from the users is attributes of interest. According to users' specification, database is scrutinized to produce a working subset that only contains the attributes of interest while the rest are excluded. With this dataset, the Apriori procedure searches for frequent large

itemsets. The advantage is apparent since irrelevant records are filtered out, the result is more meaningful and the running time is also reduced.

We summarize the above surveyed research systems in Table 0.2.. Each system is characterized by 1) its domain of application, 2) type of domain knowledge employed, 3) usage of domain knowledge, and 4) data mining techniques that are adapted to incorporate the domain knowledge.

System	Problem domain	Type of domain knowledge	Usage of domain knowledge	Data mining method
Daniels *et al.* [8]	Business Intelligence	Monotonicity constraints	modify mining algorithms to embody the knowledge directly	Decision tree and neural network
Ambrosino *et al.* [2]	Medical decision	Attribute-relation, interpretation of result	Experts interact directly with mining in both pre– and post– processing stages	Decision tree
Pazzani *et al.* [1]	Predicate learning	Taxonomy, attribute-relation rules, attribute correlations	Preprocessing data	FOCL
Sinha *et al.* [3]	Business Intelligence	Expert rules	Rule's prediction cascaded as an input to classifier	Seven typical classification algorithms
Yoon *et al.* [9]	Query optimization	Taxonomy, attribute relation rules and correlation	Transform data mining queries	Not specified
Hirsh *et al.* [4]	DNA sequence classification	Attribute relation rules	Forming new set of attributes	C4.5 and neural network
Vikram *et al.* [10]	Association rule mining	Attribute of interest	Preprocessing data	Association rules
Weiss *et al.* [7]	Consultation	Question-answer pairs derived from interviewing experts	Question-answer pairs serve as part of the input to a mining system	No restriction
Kopanas *et al.* [6]	Business intelligence	Comprehensive information pertaining to a domain	For each stage of mining, discussing the use of certain type of domain knowledge in general	No restriction

TABLE 0.1. Summary of systems that employ domain knowledge.

Staab and Hotho [11] describe an ontology-based text clustering approach. They develop a preprocessing method, called COSA, one of the earliest to utilize the idea of mapping terms in the text to concept in the ontology. The authors point out that the size of the high-dimensional term vector representation of the text document is the principal problem faced by previous algorithms. By mapping terms to concepts, it essentially aggregates terms and reduces the dimensionality.

The mapping of terms to concepts can be also seen as a process of semantic annotation. It is realized in COSA by using some shallow and efficient natural language processing tools. After the mapping process, COSA further reduces the dimensionality by aggregating concepts using the concept heterarchy defined in the "core ontology" used in their framework. The idea is navigating the heterarchy top-down splitting the concepts with most support (number of mapping terms) into their sub-concepts and abandoning the concepts with least support. The rationale is that too (in-) frequent concept occurrences are not appropriate for clustering. Note that the definition of a "core ontology" in this paper was developed prior to the emergence of OWL. The concept heterarchy should be thought of as equivalent to the subsumption hierarchy (taxonomy) in OWL. Despite the out-dated definition of terminology, COSA pioneers in incorporating ontology in text clustering and displays some generality over the confines of any specific domain.

Wen *et al.* [12] devise a framework that solves the genomic information retrieval problem by using ontology-based text clustering. The core idea is an extension to COSA. Documents containing genomic information are first annotated based on UMLS so that the terms are mapped to concepts. Then the authors point out that even the dimension of clustering space is dramatically reduced, there still exists the problem that a document is often full of class-independent "general" words and short of class-specific "core" words, which leads to the difficulty of document clustering because class-independent words are considered as noise in clustering. To solve this problem, the authors propose a technique for concept frequency

re-weighing which takes into consideration the concept subsumption hierarchy defined in the domain ontology. Finally, from the re-weighed concept vector representation, a cluster language model can be generated for information retrieval.

Fang *et al.* [13] propose an ontology-based web documents classification and ranking method. The contribution of this work is that they introduce a way to automatically augment or tailor the existing ontology to fit the specific purpose, while in previous work one has to either manually create an ontology from scratch or adopt some well established domain ontology. Their technique is to enrich a certain ontology using terms observed in the text document. This is done with the help of WordNet. Specifically, for example, if the sense of a term appears to be a synonym of the sense of a concept according to WordNet, the term will be added to the ontology as a sibling of the concept. The enriched ontology is then treated as a representation of the category to which some text document is classified. The proposed classification is done by simply comparing the similarity between ontologies and the term vector representing the text document. This implies that first, multiple ontologies should be provided for choice, and second, for each category of the corpus there should be one corresponding ontology. These assumptions appear cumbersome though the authors point to the Open Directory Project as a source for initial ontologies in their experiment. Moreover, this process does not fit into traditional classification as there is no training phase. It is more similar to clustering with known number of clusters.

Cheng *et al.* [14] studied document clustering problem as a means to efficient knowledge management. They utilized ontologies to overcome the ambiguity problem frequently seen in natural language since "an ontology includes a selection of specific sets of vocabulary for domain knowledge model construction, and the context of each vocabulary is represented and constrained by the ontology." They developed a system called Ontology-based Semantic Classification (OSC) Framework that consists of two main components: Content-based Free Text Interpreter (CFTI) and Context-based Categorization Agent (CCA). CFTI leverages

on the Link Grammar capability for syntactical analysis of a sentence. At the same time, the lexical meaning analysis of a sentence is supported through the integration with ontological models such as the WordNet. The context models produced from CFTI correlate the content of a particular document with the context of the user. The role of the CCA is to further enhance the usability of these context models by classifying them according to the user interest. The OSC framework seems appealing but the authors did not provide any implementation details nor experiment results. It was more of a research proposal and it would be interesting to see the performance of the system when the authors make the proposal a reality.

Taghva *et al.* [15] reported on the construction of an ontology that applies rules for identification of features to be used for an email classification system, called "Ecdysis". The ontology is designed for the purpose of encoding expert rules deciding the email category. Therefore it contains only those concepts that are aspects of such rules. CLIPS is used to implement rules and the inference with rules is based on a "match-and-fire" mechanism: One or more features of an email instance would be matched with instances of classes from the ontology. If there was a successful match, then the rule would fire, causing the email to have some certain feature. This feature becomes one of many that can now be used for training and classification with our Bayesian classifier Ecdysis. The authors claim that preliminary tests show that these additional features enhance the accuracy of the classification system dramatically.

Tenenboim *et al.* [16] proposed an automatic method for classification of news using hierarchical news ontology. The system they developed was called ePaper. It is designed to aggregate news items from various news providers and delivers to each subscribed user a personalized electronic newspaper, utilizing content-based and collaborative filtering methods. The ePaper can also provide users "standard" (*i.e.*, not personalized) editions of selected newspapers, as well as browsing capabilities in the repository of news items. The

classification task performed in the ePaper system aims at classifying each incoming news document to one or several concepts in the news Ontology. In this sense, only the target classes in the classification process are annotated by ontological terms. Since the users' profiles are also defined using the same set of ontological terms, a content-based filter is able to compare the similarity between a user's profile and classified categories of news. Based on results of the classifier and content-based filter, the personalization engine of the system is able to provide a personalized paper.

Lula *et al.* [17] proposed an ontology-based cluster analysis framework. They discussed various aspects of similarity measure between objects and sets objects in ontology-based environment. They devised an ontology-based aggregation function to calculate similarity between two objects which takes into account taxonomy similarity, relationship similarity and attribute similarity. For example, path distance, Jaccard coefficient and measures based on information theory can be used to calculate taxonomy similarity. Relationship similarity can be determined by calculating similarity of objects that participate in the relationship. Attribute similarity is determined by comparing values of the attributes. The authors claim that the framework with ontology-based similarity measure opens the possibility for various clustering application. But apparently much work still remains. It is unclear how the aggregation function is defined though each of its components can be solved separately. A proper aggregation is highly possible to be application-specific, which may suggest the need of a learning framework to derive such function.

Li *et al.* [18] developed a new decentralized P2P architecture-ontology-based community overlays. The system exploits the semantic property of the content in the network to cluster nodes sharing similar interest together to improve the query and searching performance. To do that, they proposed a query routing approach that organizes nodes into community overlays according to different categories defined in nodes' content ontology. A community overlay is composed of nodes with similar interest. Queries are only forwarded to semantically

related overlays, thus alleviating the traffic load. According to taxonomic information in the ontology, peers (nodes) can be clustered into ontological terms. The authors further discussed routing policy among communities and issues related to community overlay maintenance, which is out of scope of this paper. This study introduced a new data mining application besides text document clustering. But their principle remained the same as other surveyed work: ontology is used as an abstraction to data. By doing so, some performance metrics of the data mining task can be improved.

Adryan *et al.* [19] developed a system called GO-Cluster which uses the tree structure of the Gene Ontology database as a framework for numerical clustering, and thus allowing a simple visualization of gene expression data at various levels of the ontology tree. Shen *et al.* [20] proposed a new method of association rules retrieval that is based on ontology and Semantic Web. They argue that ontology-based association rules retrieval method can well deal with the problems of rule semantics sharing, rule semantics consistency and intelligibility.

System	Ontology construction	Annotation method	Type of sources	Data mining method
Staab *et al.* (COSA) [11]	Manual creation	Shallow NLP method	Text	Clustering based on "bag-of-concept" representation plus concept aggregation
Wen *et al.* [12]	Off-the-shelf (UMLS)	Manual	Text	Clustering based on "bag-of-concept" representation plus concept frequency reweighing
Fang *et al.* [13]	Manual creation of "core" ontology and update on the fly	Manual	Text	Clustering based on "bag-of-concept" representation plus feed back to enrich ontology
Cheng *et al.* (OSC) [14]	Off-the-shelf (WordNet)	Rule-based NLP	Text	Not specified
Taghva *et al.* (Ecdysis) [15]	Manually creation, incorporated with a rule inference system	Manual	Email / text	Classification with additional features derived from rules
Tenenboim *et al.* [16]	Manual creation	Manual	News archive /text	Not specified
Lula *et al.* [17]	Not specified	Manual	Text	Hierarchical agglomerative clustering
Li *et al.* [18]	Off-the-shelf (Open Directory Project)	Manual	P2P user / resource profile data	Not specified
Adryan *et al.* [19]	Off-the-shelf (Gene Ontology)	Manual	Gene expressions	Hierarchical clustering with instance regrouping based on GO annotation

TABLE 0.2. Summary of ontology-based data mining systems.

aabc

0.2. Graph-based Approach for Knowledge Representation

Graph-based approaches for representing knowledge have long been used in philosophy, psychology, and linguistics. The computer counterpart to this means is the so-called *semantic network* that represents knowledge in patterns of interconnected nodes and arcs which were first developed for artificial intelligence and machine translation.

The semantic network, and graph-based approaches for KR in general, are motivated by the desirable qualities of graph for both modeling and computation. From a modeling viewpoint, basic graphs are easily understandable by users, and it is always possible to split up a large graph into smaller ones while keeping its semantics. From the computational viewpoint, graph is one of the most studied objects in mathematics. Considering graphs instead of logical formulas provides another view of knowledge constructs (*e.g.*, some notions like path, cycle, or connected components are natural on graphs) and provides insights to algorithmic ideas [21]. In light of these motivations, what is common to all semantic networks is a declarative graphic representation that can be used either to represent knowledge or to support automated systems for reasoning about knowledge.

According to Sowa [22], following are six of the most common kinds of semantic networks.

1. Definitional networks focus on the is-a or subtype relation among concepts. The resulting network, also called a generalization or subsumption hierarchy, supports the rule of inheritance to propagate properties from a supertype to all of its subtypes. The information in these networks is often assumed to be necessarily true.

2. Assertional networks are designed to assert propositions. Unlike definitional networks, the information in an assertional network is assumed to be contingently true, unless

it is explicitly marked with a modal operator. Some assertional networks have been proposed as models of the conceptual structures underlying natural language semantics.

3. Implicational networks use implication as the primary relation for connecting nodes. They may be used to represent patterns of beliefs, causality, or inferences.

4. Executable networks include some mechanism, such as marker passing or attached procedures, which can perform inferences, pass messages, or search for patterns and associations.

5. Learning networks build or extend their representations by acquiring knowledge from examples. The new knowledge may change the old network by adding and deleting nodes and arcs or by modifying numerical values, called weights, associated with the nodes and arcs.

6. Hybrid networks combine two or more of the previous techniques, either in a single network or in separate, but closely interacting networks.

Among all variants of semantic networks, we emphasize the most on the usage of definitional networks to incorporate domain knowledge in data mining. The kind of knowledge that are best captured by this kind of network is subsumption hierarchy, on which a distance (similarity) measure can be reasonably defined. Such measure is essential in many data mining tasks. It is possible to extend, in a straightforward manner, data mining algorithms that depend on analyzing distances between entities in factual knowledge to work with distances between those in ontological knowledge.

In addition, one of the most prominent KR formalism families among current systems of definitional networks, description logics (DLs), formerly called terminological logics or concept languages, have been a successful attempt to combine well-defined logical semantics with efficient reasoning [23]. They are derived from an approach proposed by Woods [24]

and implemented by Brachman [25] in a system called Knowledge Language One (KL-ONE). Recent description logics are DAML+OIL [26] and its successor OWL, which are intended for representing knowledge in the semantic web [27]–a giant semantic network that spans the entire Internet.

0.2.1. Graph Representation of RDF

According to the W3C specification for RDF semantics [28], an RDF graph, or simply a graph, is defined as a set of RDF triples. A subgraph of an RDF graph is a subset of the triples in the graph. A triple is identified with the singleton set containing it, so that each triple in a graph is considered to be a subgraph. A proper subgraph is a proper subset of the triples in the graph. A ground RDF graph is one with no blank nodes.

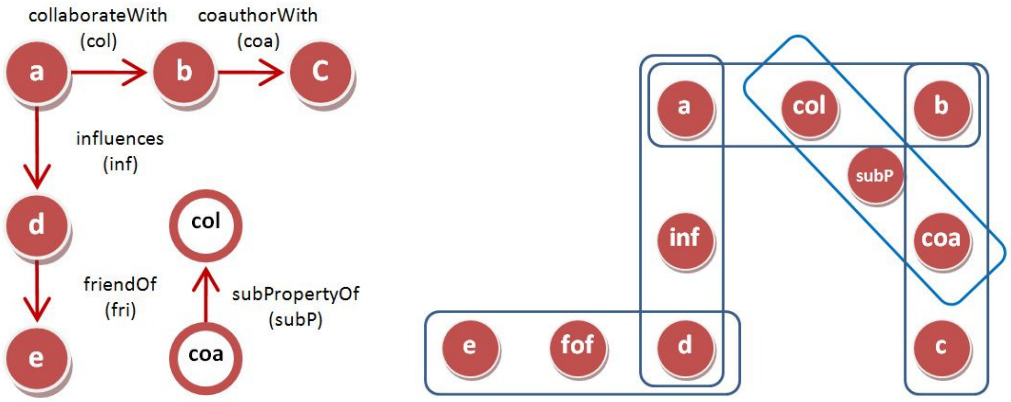
RDF triples can be visualized as a *directed labeled graph*,

$$\boxed{\text{subject}} \xrightarrow{predicate} \boxed{\text{object}},$$

where subjects and objects are represented as nodes, and predicates as edges. The directed labeled graph model for RDF is straightforward and convenient in most cases. But inconsistency arises when using triples to make assertions on predicates. The directed labeled graph model of RDF makes the artificial distinction between resources and properties. The results of the understanding of RDF bounded by the this model becomes especially evident in the limitations of current RDF query languages as studied in [29].

The following example illustrates this point of view.

In this example, two different levels of information are expressed. At the data level, the following statements describe the relationship between nodes representing people: $\langle a\ col\ b\rangle$, $\langle b\ coa\ c\rangle$, $\langle a\ inf\ d\rangle$ and $\langle d\ fri\ e\rangle$. At the schema level, another statement asserting that *coa* is a sub-property of *col*: $\langle coa\ subP\ col\rangle$. In this case, *col* and *coa* become both nodes and



(A) (B)

FIGURE 0.1. An example of nodes connected by different links and some relationship between the links, represented by A) a simple graph, and B) a hypergraph.

edges in the directed labeled graph, causing the inconsistency (as depicted in Fig. 0.1.(A). One way to overcome this is to model RDF as a hypergraph.

A hypergraph [30] is a generalization of a traditional graph where edges, called hyperedges, can connect more than two vertices. If each edge in a hypergraph covers the same number of nodes, it's called r-uniform hypergraph, r being the number of nodes on each edge. Any RDF Graph can be represented by a simple ordered 3-uniform hypergraph, in which an RDF triple corresponds to a hypergraph edge, the nodes being the subject, predicate and object in this order. In this way, both meta-data and data level statements can be integrated in a consistent model. In Fig. 0.1.(b), the information in the above example is represented using a hypergraph.

Formally, a hypergraph $G = (V, E)$, is a pair in which V is the vertex set and E is the hyperedge set where each $e \in E$ is a subset of V. A weighted hypergraph is a hypergraph that has a positive number $w(e)$ associated with each hyperedge e; called the weight of hyperedge e: Denote a weighted hypergraph by $G = (V, E, w)$. The degree of a vertex $v \in V$, $d(v)$, is defined as $d(v) = \sum_{v \in V, e \in E} w(e)$. The degree of a hyperedge e, denoted as $\delta(e)$, is the number of vertices in e, i.e. $\delta(e) = |e|$. A hyperedge e is said to be incident with a vertex v

when $v \in e$. The hypergraph incidence matrix $\mathbf{H} \in \mathbb{R}^{|V| \times |E|}$ is defined as

$$h(v,e) = \begin{cases} 1, & v \in e \\ 0, & otherwise \end{cases}$$

Throughout the rest of the paper, the diagonal matrix forms for $\delta(e)$, $w(e)$, $d(v)$ are denoted as $\mathbf{D}_e$, $\mathbf{W} \in \mathbb{R}^{|E|}$, and $\mathbf{D}_v \in \mathbb{Z}^{|V|}$, respectively.

0.3. Graphs in Data Mining

0.3.1. Graph Representation of Relational Structure

An object set endowed with pairwise relationships can be naturally illustrated as a graph in which vertices represent objects, and any two vertices that have some kind of relationship are joined together by an edge. In the case of frequent itemset mining, a set of objects with the co-occurrence relationship can be represented as directed or undirected graphs. For illustrating this point of view, let us consider a relational table depicted in Figure 0.2.(a). One can construct an undirected graph where the set of vertices is the set of relational attributes (column items) and an edge joins two vertices if the they co-occur in a tuple (as illustrated in Figure 0.2.(b)). This graph is called *Gaifman graph* [31] of a relational structure. The undirected graph can be further enriched by assigning to each edge a weight equal to the support of the 2-itemset consisting of vertices incident to the edge. Cliques (complete subgraphs) in the Gaifman graph, or *Gaifman cliques* for short, are of particular interest because every tuple (ground atom) in data corresponds to a Gaifman clique. However, ambiguity arises as not all Gaifman cliques have matching tuple in the data. There exists cases where cliques are incidental in the sense that several relational ground atoms play together to induce a clique configuration in the Gaifman graph, but no

ground atom covers the entire clique (e.g., the clique of $\{A, B, C, D\}$ in Figure 0.2.(b) does not correspond to any tuple in the relational table).

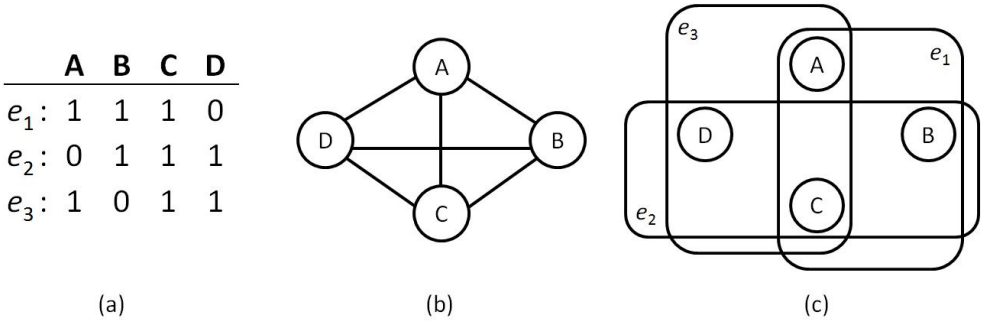


FIGURE 0.2. (a) an example transaction table; (b) the Gaifman graph representation of the table; (c) The hypergraph representation of the table

A natural way to remedy the ambiguity is to represent the relational data as a *hypergraph* [30]. A hypergraph is a generalization of traditional graph. An edge in the hypergraph, called hyperedge, can connect more than two vertices. In other words, every hyperedge is an arbitrary nonempty subset of vertices. It is obvious that a simple graph is a special kind of hypergraph with each hyperedge containing only two vertices. In this paper, we propose to employ hypergraphs to model relational structure for finding semantically associated itemsets. Specifically, we propose to construct a hyperedge for each tuple. The relational attributes constitute the universe of vertices in the hypergraph. In this representation, each hyperedge has an exact one-to-one correspondent tuple (see Figure 0.2.(c), for example).

0.3.2. Similarity Measure

Given graph-based representation of information sources, meaningful similarity measure s between nodes in the graph is critical in numerous data mining tasks. Take the simple network in Fig. 0.1.(B) for example, suppose given a task of friend recommendation based

on the information in this graph, the interesting question is whether c or e is a better choice of recommendation to a. To answer this question, it is natural to compare the similarity measures $s(a, c)$ and $s(a, e)$. In a rough sense, on can identify in the hypergraph representation that there are two paths between a and c (the formal definition for paths in hypergraphs will be given in Section **??**), while only one between a and e. It's intuitive to conclude that a and c are more similar, or closer, than a and e. This gives us a hint that meaningful similarity measures on the graph should satisfy the following intuitions:

1. The more paths connecting two nodes, the closer they are.

2. The shorter the paths, the closer they are.

In other words, the more "short" connections between two given nodes, the more similar those nodes are. To this end, we propose to employ the following quantities as the candidate similarity measure since both of them have the desired property. They are, namely, the *commute time distance* based similarity measure from the random walk model on hypergraph, and the inner product similarity based on the *pseudoinverse of the hypergraph Laplacian*. They are all based on the random walk model on hypergraph. In the following, we briefly introduce the theory of random walk.

0.3.2.1. Random Walk

Random Walk on Simple Graph Given a graph and a starting point we select a neighbor of it at random and move to this neighbor then we select a neighbor of this point at random and move to it etc. The random sequence of points selected this way is a random walk on the graph. In other words, a random walker can jump from vertex to vertex and each vertex therefore represents a state of the Markov chain. The average first-passage time $m(k|i)$ [32] is the average number of steps needed by a random walker for reaching state k for the first time, when starting from state i. The symmetrized quantity $n(i, j) = m(j|i)+m(i|j)$

called the average commute time [32], provides a distance measure between any pair of states. The fact that this quantity is indeed a distance on a graph was proved independently by Klein and Randic [33] and Gobel and Jagers [34].

The Laplacian matrix $\mathbf{L}$ of a graph is widely used for finding many properties of the graphs in spectral graph theory. Given node degree matrix $\mathbf{D}$ and graph adjacency matrix $\mathbf{A}$, the Laplacian matrix of the graph is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The normalized Laplacian is given by $\mathbf{L}_N = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{I}$ is the identity matrix. The average commute time $n(i, j)$ can be computed in closed form from the Moore-Penrose pseudoinverse of $\mathbf{L}$ [35], denoted by $\mathbf{L}^+$.

Various quantities derived from random walk on graph has been used in a number of applications. Fouss et al. [36] compared twelve scoring algorithms based on graph representation of the database to perform collaborative movie recommendation. Pan et al. [37] developed a similarity measure based on random walk steady state probability to discover correlation between multimedia objects containing data of various modalities. Yen et al. [38] introduced a new k-means clustering algorithm utilizing the random walk average commute time distance. Zhou et al. [39] presented a unified framework based on neighborhood random walk to integrate structural and attribute similarities for graph clustering.

REFERENCES CITED

[1] Michael Pazzani and Dennis Kibler. The Utility of Knowledge in Inductive Learning. *Mach. Learn.*, 9:57–94, June 1992. ISSN 0885-6125.

[2] R. Ambrosino and B.G. Buchanan. The use of physician domain knowledge to improve the learning of rule-based models for decision-support. In *Proceedings of the Annual Fall Symposium of the American Medical Informatics Association*, pages 192–196, 1999.

[3] Atish P. Sinha and Huimin Zhao. Incorporating Domain Knowledge Into Data Mining Classifiers: An Application In Indirect Lending. *Decis. Support Syst.*, 46:287–299, December 2008. ISSN 0167-9236.

[4] Hayam Hirsh and Michiel Noordewier. Using Background Knowledge to Improve Inductive Learning: A Case Study in Molecular Biology. *IEEE Expert: Intelligent Systems and Their Applications*, 9:3–6, October 1994. ISSN 0885-9000.

[5] Carsten Pohle. Integrating and Updating Domain Knowledge with Data Mining. In *VLDB PhD Workshop*, 2003.

[6] Ioannis Kopanas, Nikolaos M. Avouris, and Sophia Daskalaki. The Role of Domain Knowledge in a Large Scale Data Mining Project. In *Proceedings of the Second Hellenic Conference on AI: Methods and Applications of Artificial Intelligence*, SETN '02, pages 288–299, London, UK, 2002. Springer-Verlag. ISBN 3-540-43472-0.

[7] Gary Weiss and Foster Provost. The Effect of Class Distribution on Classifier Learning: An Empirical Study. Technical report, Department of Computer Science, Rutgers University, 2001.

[8] Hennie Daniels, Ad Feelders, and Marina Velikova. Integrating Economic Knowledge in Data Mining Algorithms. Discussion Paper 2001-63, Tilburg University, Center for Economic Research, 2001.

[9] Suk-Chung Yoon, Lawrence J. Henschen, E. K. Park, and Sam Makki. Using domain knowledge in knowledge discovery. In *Proceedings of the eighth international conference on Information and knowledge management*, CIKM '99, pages 243–250, New York, NY, USA, 1999. ACM. ISBN 1-58113-146-1.

[10] Vikram Singh and Sapna Nagpal. Integrating User's Domain Knowledge with Association Rule Mining. *CoRR*, abs/1004.3568, 2010.

[11] Steffen Staab and Andreas Hotho. Ontology-based text document clustering. In *Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'03 Conference held in Zakopane*, pages 451–452, 2003. ISBN 3-540-00843-8.

[12] *Ontology Based Clustering for Improving Genomic IR*, 2007.

[13] Jun Fang, Lei Guo, XiaoDong Wang, and Ning Yang. Ontology-based automatic classification and ranking for web documents. In *Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery - Volume 03*, pages 627–631, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2874-0.

[14] Ching Kang Cheng, Xiaoshan Pan, and Franz J. Kurfess. Ontology-Based Semantic Classification of Unstructured Documents. In Andreas Nürnberger and Marcin Detyniecki, editors, *Adaptive Multimedia Retrieval*, volume 3094 of *Lecture Notes in Computer Science*, pages 120–131. Springer, 2003. ISBN 3-540-22163-8.

[15] Kazem Taghva, Julie Borsack, Jeffrey Coombs, Allen Condit, Steve Lumos, and Tom Nartker. Ontology-based Classification of Email. In *Proceedings of the International Conference on Information Technology: Computers and Communications*, ITCC '03, pages 194–, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1916-4.

[16] L. Tenenboim, B. Shapira, and P. Shoval. Ontology-based Classification of News in an Electronic Newspaper. In *Proceedings of INFOS 2008*, pages 89–97, Varna, Bulgaria, 2008.

[17] PawełLula and Grażyna Paliwoda-Pekosz. An ontology-based cluster analysis framework. In *Proceedings of the first international workshop on Ontology-supported business intelligence*, OBI '08, pages 7:1–7:6, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-219-1.

[18] Juan Li and Son Vuong. Ontology-Based Clustering and Routing in Peer-to-Peer Networks. In *Proceedings of the Sixth International Conference on Parallel and Distributed Computing Applications and Technologies*, PDCAT '05, pages 791–795, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2405-2.

[19] Boris Adryan and Reinhard Schuh. Gene-Ontology-based clustering of gene expression data. *Bioinformatics*, 20:2851–2852, November 2004. ISSN 1367-4803.

[20] Bin Shen, Min Yao, Zhaohui Wu, Yangu Zhang, and Wensheng Yi. Ontology-based Association Rules Retrieval using Protege Tools. In *Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, pages 765–769, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2702-7.

[21] Michel Chein and Marie-Laure Mugnier. *Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs*. Springer, London, 2008. ISBN 978-1-84800-285-2.

[22] John F. Sowa. *Semantic Networks*. Wiley, 1987.

[23] John F. Sowa. Principles of semantic networks. Morgan Kaufmann, 1991.

[24] William A. Woods. What's in a Link: Foundations for Semantic Networks. In Daniel G. Bobrow and Allan M. Collins, editors, *Representation and Understanding: Studies in Cognitive Science*, pages 35–82. Academic Press, 1975.

[25] Ronald J. Brachman, Deborah L. McGuinness, Peter F. Patel-Schneider, Lori A. Resnick, Lori Alperin Resnick, and Alexander Borgida. Living with CLASSIC: When and How to Use a KL-ONE-Like Language. In *Principles of Semantic Networks*, pages 401–456. Morgan Kaufmann, 1991.

[26] Ian Horrocks. Daml+oil: a description logic for the semantic web. *IEEE Data Engineering Bulletin*, 25:4–9, 2002.

[27] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web: Scientific American. *Scientific American*, May 2001.

[28] RDF semantics, W3C recommendation, 2004. URL `http://www.w3.org/TR/rdf-mt/`.

[29] Renzo Angles, Claudio Gutierrez, and Jonathan Hayes. Rdf query languages need support for graph properties. Technical report, 2004.

[30] C. Berge. Hypergraphs. *Bull. Symbolic Logic*, 1989.

[31] Ian Hodkinson and Martin Otto. Finite conformal hypergraph covers and Gaifman cliques in finite structures. *Bull. Symbolic Logic*, 9:387–405, 2002.

[32] L. Lovasz. Random Walks on Graphs: A Survey. In *Combinatorics*, pages 353–397, Budapest, 1993. Janos Bolyai Math. Soc.

[33] D.J. Klein and M. Randic. Resistance Distance. *J. Math. Chemistry*, 12:81–95, 1993.

[34] F. Gobel and A. Jagers. Random Walks on Graphs. *Stochastic Processes and Their Applications*, 2:311–336, 1974.

[35] S. Barnett, editor. *Matrices: Methods and Applications*. Oxford Univ. Press, 1992.

[36] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. Random-Walk Computation Of Similarities Between Nodes Of A Graph, With Application To Collaborative Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 19:2007, 2006.

[37] Jia-Yu Pan, Hyungjeong Yang, Christos Faloutsos, and Pinar Duygulu. Cross-modal Correlation Mining Using Graph Algorithms. *Knowledge Discovery and Data Mining: Challenges and Realities with Real World Data.*, 2006.

[38] L. Yen, L. Vanvyve, D. Wouters, F. Fouss, F. Verleysen, and M. Saerens. Clustering Using A Random-Walk Based Distance Measure. In *Proceedings of ESANN'2005*, 2005.

[39] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. Graph Clustering Based On Structural/Attribute Similarities. *Proc. VLDB Endow.*, 2:718–729, August 2009. ISSN 2150-8097.