A General Strategy for Knowledge Acquisition from Semantically Heterogeneous Data Sources

Doina Caragea  
Department of Computing and Information Sciences  
Kansas State University  
234 Nichols Hall, Manhattan, KS 66506

Jie Bao and Vasant Honavar  
Artificial Intelligence Research Lab  
Department of Computer Science  
Iowa State University  
226 Atanasoff Hall, Ames, IA 50011

Abstract

With the advent of the Semantic Web, there is increased availability of meta data (ontologies) that make explicit the semantic commitments associated with the data sources. Together with tools for specifying mappings between ontologies, this has opened up for the first time, the possibility of acquiring knowledge from such ontology extended, semantically disparate data sources. Hence, there is an urgent need for machine learning algorithms for building predictive models (e.g., classifiers) in a setting where there is no unique global interpretation of data from semantically disparate sources and it is neither feasible nor desirable to collect data from such sources in a centralized data warehouse. We formulate the problem of learning classifiers from a set of related, semantically heterogeneous data sources, under the assumption that ontologies and mappings from a user ontology to the data source ontologies are given. We design a general strategy for learning classifiers from such data sources by reducing the problem of learning to the problem of answering queries from semantically heterogeneous data and we show how to answer such queries.

Introduction

The availability of large amounts of data in many application domains offer unprecedented opportunities in computer assisted data-driven knowledge acquisition in a number of applications including, in particular, data-driven scientific discovery (e-Science) in bioinformatics, environmental informatics, geo-informatics, neuro-informatics, health informatics, etc. or data-driven decision making in business and commerce (e-Business and e-commerce).

Machine learning techniques (Mitchell 1997; Duda, Hart, & Stork 2000), in addition to traditional statistical techniques (Casella & Berger 2001), offer some of the most cost-effective approaches to analyzing, exploring and extracting knowledge (features, correlations, and other complex relationships and hypotheses that describe potentially interesting regularities) from such data sources. However, the applicability of current knowledge acquisition techniques is challenged by the nature and the scale of the data available. More precisely:

(a) Data repositories are large in size, dynamic, and physically distributed. Consequently, it is neither desirable nor feasible to gather all of the data in a centralized location for analysis. Hence, there is a need for efficient algorithms for analyzing and exploring multiple distributed data sources without transmitting large amounts of data.

(b) Autonomously developed and operated data sources often differ in their structure and organization (e.g., relational databases, flat files, etc.) and the operations that can be performed on the data sources (e.g., types of queries - relational queries, statistical queries, keyword matches). Hence, there is a need for theoretically well-founded strategies for efficiently obtaining the information needed for analysis within the operational constraints imposed by the data sources.

(c) Autonomously developed data sources are semantically heterogeneous. The ontological commitments associated with a data source (and hence its implied semantics) are typically determined by the data source designers, based on their understanding of the intended use of the data. Very often, data sources that are created for use in one context or application find use in other contexts or applications, and therefore, semantic differences among autonomously designed, owned, and operated data repositories are simply unavoidable. Therefore, effective use of multiple sources of data in a given context requires reconciliation of semantic differences. As a consequence, there has been significant community-wide efforts aimed at the construction of ontologies (e.g., Gene Ontology - GO1 - in biology, Unified Medical Language System - UMLS2 - in heath informatics, Semantic Web for Earth and Environmental Terminology - SWEET 3 - in geospatial informatics). However, collaborative scientific discovery applications often require users to be able to analyze data from various perspectives. There is no single privileged perspective that can serve all users, or for that matter, even a single user, in every context. Hence, there is a need for methods that can dynamically and efficiently extract and integrate information needed for knowledge acquisition, from semantically heterogeneous data, from a user’s

---

1www.geneontology.org  
2www.nlm.nih.gov/research/umls  
3sweet.jpl.nasa.gov
perspective, based on user-specified ontologies and user-specified mappings between ontologies.

Against this background, we note that a large class of data sources on the Semantic Web can be viewed (at a certain level of abstraction) as a collection of semantically disparate relational data sources that are semantically related, from a user’s point of view, in the context of a specific knowledge acquisition task. We design a general strategy for learning classifiers from multiple semantically disparate, geographically distributed, relational data sources on the Semantic Web.

The rest of the paper is organized as follows: We first introduce the concepts and definitions needed to formulate the problem addressed. Next, we formulate the problem of learning from semantically heterogeneous data and describe a general strategy for transforming algorithms for learning classifiers from relational data into algorithms for learning classifiers from semantically disparate, relational data sources, using ontologies and mappings between ontologies, in a setting where it is neither feasible nor desirable to integrate all the data available into a single relational data warehouse. We conclude with a summary and a brief discussion of related work.

Concepts and Definitions

Ontology-Extended Data Sources and User Views

We define an ontology-extended relational data source (OERDS) as a tuple \( D = \{ D, S, O \} \), where \( D \) is the actual data set in the data source, \( S \) represents the data source schema and \( O \) represents the data source ontology (Bonatti, Deng, & Subrahmanian 2003; Caragea et al. 2005b).

In the relational model, each data source consists of a set of concepts \( X_1, \ldots, X_n \) and a set of properties of these concepts \( P_1, \ldots, P_m \). Each concept has associated with it, a set of attributes denoted by \( A(X_i) \) and a set of \( k \)-ary relations \( (k > 1) \) denoted by \( R(X_i) \). Each attribute \( A_i \) takes values in a set \( V(A_i) \). The concepts and the properties of the concepts (attributes and relations) define the schema of a relational data source. A data set \( D \) is an instantiation \( I(S) \) of a schema \( S \) (Getoor et al. 2001).

The ontology \( O \) of an OERDS \( D \) consists of two parts: structure ontology, \( O_S \), that defines the semantics of the data source schema (concepts and properties of the concepts that appear in data source schema \( S \)); and content ontology, \( O_I \), that defines the semantics of the content of data (values and relationships between values that the attributes can take in instantiations of schema \( S \)). Isa relationships induce schema concept hierarchies (SCHs) over subsets of concepts in a schema and attribute value hierarchies (AVHs) over values of attributes (AVHs can be seen as defining a type hierarchy over the corresponding attributes). Thus, an ontology \( O \) can be decomposed into a set of schema concept hierarchies \( \{ C_1, \ldots, C_r \} \) and a set of attribute value hierarchies \( \{ T_1, \ldots, T_l \} \), with respect to the isa relationship. A cut (or level of abstraction) through an SCH or AVH induces a partition of the set of leaves in that hierarchy. A global cut through an ontology consists of a set of cuts, one for each constituent hierarchy.

On the Semantic Web, it is unrealistic to assume the existence of a single global ontology that corresponds to a universally agreed upon set of ontological commitments for all users. Instead, it is much more realistic to allow each user or a community of users to choose the ontological commitments that they deem useful in a specific context. A user ontology \( O_U \), together with a set of interoperation constraints \( IC \), and the associated set of mappings \( \{ \psi_i | i = 1, p \} \) from the user ontology \( O_U \) to the data source ontologies \( O_1 \cdot \cdot \cdot O_p \) define a user view (Caragea et al. 2005b). In the relational setting considered in this paper, the interoperation constraints can be equality constraints or inclusion constraints and can be defined at the concept level (between related concepts), property level (between related attributes or relations) and at the attribute value level (between related attribute values).

Bibliography Example

We will use an example from the bibliography domain to illustrate the main notions introduced above. Consider the problem of classifying computer science research papers into categories from a topic hierarchy (e.g., Artificial Intelligence, Networking, Data Mining, Relational Data Mining, etc.) (McCallum et al. 2000). A user interested in a document classification task, might consider using several data sources, such as CiteULike (http://www.citeulike.org/) from UMBC; the Collection of Computer Science Bibliographies4 from University of Karlsruhe; MIT Libraries (http://libraries.mit.edu/index.html), INRIA Reference Database (http://ontoweb.org/), etc., for learning classifiers. The Ontology Alignment Evaluation Initiative (OAEI) has made available a Test Library (http://oaei.inrialpes.fr/2005/benchmarks/) that contains representative ontologies for the data sources above. In this case, the structure ontologies define the relevant concepts, such as Reference, Book, Article, Journal, Conference, etc.) relationships between classes (see Figure 2 for class hierarchies contained in the INRIA, MIT and a user ontology, respectively), and properties of the concepts such as Article author; Author article; Author position; Article journal; Journal, etc. The properties in these ontologies include both attributes (e.g., position) and binary relations (e.g., author).

Figure 1 shows a small fragment of the schema ontology corresponding to a user view of a reference data source, using standard entity-relationship (ER) notation. Figure 2 identifies fragments of the SCHs associated with related subsets of concepts in INRIA, MIT and user schema ontologies. Figure 3 shows concept level interoperation constraints (equality = and inclusion <) between the user SCH and the MIT and INRIA SCHs: \( x = y \) means that \( x \) and \( y \) are equivalent, \( x < y \) means that \( y \) subsumes \( x \), i.e., \( y \) is more general than \( x \).

The set of classes \{Part, Informal, Composite\} determines a cut through the INRIA class hierarchy.

---

4http://liinwww.ira.uka.de/bibliography/
Figure 1: Small fragment of the schema ontology corresponding to a user view, using standard ER notation (rectangles represent concepts; circles represent attributes; triangles and diamonds represent isa or arbitrary relationships among concepts, respectively).

Figure 2: Small fragments of the SCHs corresponding to INRIA and MIT data sources and to a user, using standard ER notation (with attributes being omitted to avoid cluttering the figure). The set \{Book, Misc, Part\} determines a cut in the user hierarchy. Similar cuts are showed for MIT and INRIA SCHs.

Figure 3: Interoperation constraints from the user to the MIT and INRIA SCHs.
The content ontologies describe the values that the attributes can take and relationships between these values. Assuming that a concept **Author** has an attribute called **position**, this attribute can be described using an AVH as shown in Figure 4. The set (**faculty, research staff, engineer, student**) represents a cut $\Gamma$ through this hierarchy. The set (**tenured, assistant professor, research staff, engineer, student**) is a refinement of the cut $\Gamma$. Thus, the content ontologies can be seen as types that the attributes can take.

The set of values \{**Faculty, Research Staff, Engineer**\} determines a cut through the AVT corresponding to the attribute **position**.

Similarly, ICs between properties of the user classes and properties of the data source classes can be defined (e.g., **position**=level or **proceedings**=**booktitle**), as well as IC’s between user attribute values and data source attribute values (e.g., **faculty**=academia and **engineer**=indstry).

### Partially Specified Data

Because different data sources might specify data at different levels of abstraction (relative to a user’s view), integration of OERDSs via mappings can result in data that is only partially specified. This can take the form of partially specified schemas (when schema concepts are partially specified) and partially specified attributes (when attribute values are partially specified).

The concept **Book** in the MIT hierarchy is under-specified (higher level of abstraction) with respect to (wrt) the concept **Monograph** in the user hierarchy, since a **Book** may be a **Monograph** or a **Proceedings**. On the other hand, a **Monograph** in the INRIA hierarchy is fully specified (same level of abstraction) wrt a **Monograph** in the user hierarchy. Furthermore, an **Article** in the INRIA hierarchy is over-specified (lower level of abstraction) wrt a **Part** in the user hierarchy, as any **Article** is a **Part** (of a journal). We say that: a schema concept $X_i$ in an SCH $C$ is partially specified (or under-specified) wrt a schema concept $X_j$ in an equivalent SCH $C'$ if $X_i > X_j$; $X_i$ is over-specified wrt $X_j$ if $X_i < X_j$; $X_i$ is fully specified wrt $X_j$ if $X_i = X_j$.

The attribute **grad** is under-specified wrt **Ph.D.**, since a **grad** may be a **Ph.D.** or a **M.S.,** but over-specified wrt **student** as every **grad** is a **student**. Furthermore, **freshman** is fully specified wrt **1st year**. We say that: an attribute value $v_i \in V(A)$ is partially specified (or under-specified) wrt an attribute value $v_j \in V(A')$ if $v_i > v_j$; $v_i$ is over-specified wrt $v_j$ if $v_i < v_j$; $v_i$ is fully-specified wrt $v_j$ if $v_i = v_j$.

Note that the problem of partially specified data (when attributes are partially specified) can be seen as a generalization of the problem of missing attribute values (Zhang et al. 2005), and hence it is possible to adapt statistical approaches for dealing with missing data (Little & Rubin 2002) to deal with partially specified data under **under appropriate assumptions**, (e.g., that the distribution of an under-specified attribute value is similar to that in a data source where the corresponding attribute is fully specified). Partially specified concepts pose additional challenges. Some approaches to handling partially specified concepts are: ignore a concept that becomes under-specified in a schema; or alternatively, use only the attributes that a concept inherits from its parents, while the rest (e.g., attributes specific to that concept that are not inherited from the parent) are treated as missing in all instances of the concept in that data source.

### Problem Formulation and Solution

#### Learning from OEDS

We assume the existence of

1. A collection of several related OERDSs $D_1 = \{D_1, S_1, O_1\}, \ldots, D_p = \{D_p, S_p, O_p\}$ for which: the schemas and the ontologies are made explicit; the instances in the data sources are labeled according to some criterion of interest to a user (e.g., topic categories).

2. A user view, consisting of a user ontology $O_U$ and a set of mappings $\psi_k$ that relate the user ontology to the data source ontologies $O_1, \ldots, O_p$. The user view implicitly specifies a user level of abstraction, corresponding to the leaf nodes of the hierarchies in $O_U$. The mappings $\psi_k$ can be specified manually by a user or semi-automatically derived.

3. A hypothesis class $H$ (e.g., Bayesian classifiers) defined over an instance space (implicitly specified by the concepts, their properties, and the associated ontologies in the domain of interest) and a performance criterion $P$ (e.g., accuracy on a classification task).

The problem of learning classifiers from a collection of related OEDSs can be simply formulated as follows: under the assumptions (1)-(3), the task of a learner is to output a hypothesis $h \in H$ that optimizes $P$, via the mappings $\{\psi_k\}$ corresponding to a user-specific set of interoperation constraints $IC$. As in (Caragea et al. 2005b), we say that an algorithm $L$, for learning from OERDSs $D_1, \ldots, D_p$, via the mappings $\{\psi_k\}$, is **exact** relative to its centralized counterpart $L_c$, if the hypothesis produced by $L_c$ (federated approach) is identical to that obtained by $L_c$ from the data warehouse $D$ constructed by integrating the data sources $D_1, \ldots, D_p$, according to the user view, via the same mappings $\psi_k$ (data warehouse approach).

The **exactness** criterion defined above assumes that it is possible, in principle, to create an integrated data warehouse in the centralized setting. However, in practice, the data sources $D_1, \ldots, D_p$ might impose access constraints $Z$ on a user $U$. For example, data source constraints might prohibit retrieval of raw data from some data sources (e.g., due to query form access limitations, memory or bandwidth limitations, privacy concerns) while allowing retrieval of answers to statistical queries (e.g., count frequency queries).

### Sufficient Statistics Based Strategy

Our approach to the problem of learning classifiers from OERDSs is a natural extension of a general strategy for transforming algorithms for learning classifiers from data in the form of a single flat table (as is customary in the case of a vast majority of standard machine learning algorithms) into algorithms for learning classifiers from a collection of horizontal or vertical fragments of the data, corresponding to partitions of rows or columns of the flat table, wherein each
Send the local results to the user and aggregate them to generate the classifier.

Translate these queries to queries expressed in the ontologies of each data source, using terms in the data source cut \( \Gamma_k \), and compute the results of the local queries \( q_k \) from each OERDS \( D_k \).

More precisely, the algorithm for learning an classifier from a set of related OERDSs works as follows:

- Select a global user cut \( \Gamma \) through the user ontology (both SCHs and AVHs). In particular, the user cut can correspond to the set of primitive values (i.e., leaves in the hierarchies).
- Apply the mappings \( \psi_k \) to find a cut \( \Gamma_k \), corresponding to each data source \( D_k \).
- Formulate statistical queries \( q \) using terms in the user cut \( \Gamma \).
- Translating these queries to queries expressed in the ontology of each data source \( D_k \), using terms in the data source cut \( \Gamma_k \), and compute the results of the local queries \( q_k \) from each OERDS \( D_k \).
- Send the local results to the user and aggregate them to compute the global result to the query \( q \).
- Generate the classifier \( h_\Gamma \) corresponding to the cut \( \Gamma \) based on the global result.

Note that if the cut \( \Gamma \) corresponds to the primitive concepts and values in the user hierarchies, the resulting classifier is exact with respect to the traditional classifier obtained, in principle, by integrating all the OERDSs \( D_1, \ldots, D_p \) into a central data warehouse \( D \) (using the same set of mappings \( \psi_k \) and the same assumptions for dealing with partially specified concepts and attribute values).

More precisely, the algorithm for learning an classifier from a set of related OERDSs works as follows:

- Select a global user cut \( \Gamma \) through the user ontology (both SCHs and AVHs). In particular, the user cut can correspond to the set of primitive values (i.e., leaves in the hierarchies).
- Apply the mappings \( \psi_k \) to find a cut \( \Gamma_k \), corresponding to each data source \( D_k \).
- Formulate statistical queries \( q \) using terms in the user cut \( \Gamma \).
- Translating these queries to queries expressed in the ontology of each data source \( D_k \), using terms in the data source cut \( \Gamma_k \), and compute the results of the local queries \( q_k \) from each OERDS \( D_k \).
- Send the local results to the user and aggregate them to compute the global result to the query \( q \).
- Generate the classifier \( h_\Gamma \) corresponding to the cut \( \Gamma \) based on the global result.

Figure 4: AVH associated with the attribute position of the concept Author. The set \{faculty, research staff, engineer, student\} represents a cut \( \Gamma \) through this hierarchy. The set \{tenured, assistant professor, research staff, engineer, student\} is a refinement of the cut \( \Gamma \).

In the case of learning classifiers from semantically disparate OERDSs, the statistics gathering component has to specify the statistics needed for learning as a query against the user view and assemble the answer to this query from OERDSs. This entails: decomposition of a posed query into sub-queries that the individual data sources can answer; translation of the sub-queries to the data source ontologies, via user-specific mappings; query answering from (possibly) partially specified data sources; composition of the partial answers into a final answer to the initial query (Figure 5).

More precisely, the algorithm for learning an classifier from a set of related OERDSs works as follows:

- Select a global user cut \( \Gamma \) through the user ontology (both SCHs and AVHs). In particular, the user cut can correspond to the set of primitive values (i.e., leaves in the hierarchies).
- Apply the mappings \( \psi_k \) to find a cut \( \Gamma_k \), corresponding to each data source \( D_k \).
- Formulate statistical queries \( q \) using terms in the user cut \( \Gamma \).
- Translating these queries to queries expressed in the ontology of each data source \( D_k \), using terms in the data source cut \( \Gamma_k \), and compute the results of the local queries \( q_k \) from each OERDS \( D_k \).
- Send the local results to the user and aggregate them to compute the global result to the query \( q \).
- Generate the classifier \( h_\Gamma \) corresponding to the cut \( \Gamma \) based on the global result.

Note that if the cut \( \Gamma \) corresponds to the primitive concepts and values in the user hierarchies, the resulting classifier is exact with respect to the traditional classifier obtained, in principle, by integrating all the OERDSs \( D_1, \ldots, D_p \) into a central data warehouse \( D \) (using the same set of mappings \( \psi_k \) and the same assumptions for dealing with partially specified concepts and attribute values). However, construction of such an integrated centralized data warehouse might require violation of data source access constraints (Z), and hence a learning strategy relying on a centralized data warehouse may not be implementable in practice. In contrast, the approach presented in this paper makes it possible to obtain the same classifier, as obtainable from an integrated centralized data warehouse, while circumventing the need for such a warehouse.

**Query Answering**

In the previous section, we have shown that the problem of learning classifiers from semantically heterogeneous data can be reduced to the problem of answering statistical queries from such data. In this section, we discuss the query answering process.

If the data content ontologies and the data structure ontologies (schemas) are specified, we can construct the usual relational queries with respect to such data sources, with operations such as, selection(\( \sigma \)), projection(\( \pi \)) and join(\( \bowtie \)). Such a query is said to be an *ontology-extended query* if the selection conditions in the query contain ontology operations, in the form of \( \sigma_{Position} \bowtie \) Faculty"(Person) \( \bowtie_{PerID} \) Author \( \bowtie_{RefIID} \) Reference. An ontology-extended query differs from a regular relational query in that it gives the specification of data needed...
Figure 5: Learning classifiers from OERDSs: each data source has an associated ontology and the user provides a user ontology and mappings from the data source ontologies to the user ontology.

not only by relational operations, but also by semantic operations from the associated ontology. Thus, processing ontology-extended queries differs from processing traditional queries in an RDBMS in several respects:

**Query Execution:** Ontology-extended queries are not directly executable in an RDBMS, because ontology operations, such as Position ≤ Faculty, are not directly supported by RDBMS. Some approaches to executing such queries in RDBMSs include exploiting datatype extension to the RDBMS and query rewriting.

The first approach is supported by multiple existing RDBMSs, such as Oracle and PostgreSQL. A user may define hierarchies as new data types, together with the ontological operators, that will be used in the RDBMS. However, applicability of this approach is limited. Due to security and privacy concerns, many data sources are not allowed to be extended with such new data types or executable operators by data users.

Hence, in this work we adopt the query rewriting approach, which allows the data source to be extended with ontologies without RDBMS modifications. We transform an ontology-extended query into an RDBMS query, wherein the query has embedded in it ontological assertions in a form that can be processed by an RDBMS. For example, Table 1 shows how to rewrite an atomic partial-order condition to an equivalent (i.e., has the same effect in a RDBMS) string datatype operations.

### Table 1: Atomic Condition Rewriting Rules for Partial Order Ontologies

<table>
<thead>
<tr>
<th>Original Condition</th>
<th>Rewritten Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A &lt; d</td>
<td>A ∈ sub(d)</td>
</tr>
<tr>
<td>A ≤ d</td>
<td>A ∈ {sub(d), d}</td>
</tr>
<tr>
<td>A = d</td>
<td>A = d</td>
</tr>
<tr>
<td>A ≠ d</td>
<td>A ∉ (d)</td>
</tr>
<tr>
<td>A &gt; d</td>
<td>A ∈ super(d)</td>
</tr>
<tr>
<td>A ≥ d</td>
<td>A ∈ {super(d), d}</td>
</tr>
</tbody>
</table>

**Query Translation:** Because in our work we need to query semantically heterogeneous data sources from a user’s (or application) perspective, queries are expressed using terms in a user ontology, making it necessary to transform such queries into equivalent queries expressed using data source ontology terms, while preserving, from the user’s point of view, the user-specified semantics of the data sources. We say that a translation is semantics-preserving if the original query and its translation specify the same result.

**Result Inverse Translation:** The data retrieved from an ontology-extended data source, as result to an ontology-extended query, is returned in a form that conforms to the data source ontology. To be understood by the user, the result has to be expressed in terms of the user ontology when it is possible. The basic strategy is to replace a data source ontology term with an equivalent or more general term in the user ontology. If there is no corresponding term in the user ontology, a data source term will be kept in the original form (i.e., in the data source ontology).

### Summary and Discussion

We have presented a strategy for learning classifiers from distributed, semantically heterogeneous data sources. Our strategy couples machine learning techniques with information integration techniques, making the process of knowledge acquisition from such sources transparent to the end user, as long as the implicit ontologies associated with the data are made explicit and mappings between a user ontology and data source ontologies are specified by domain experts (in principle, they could also be semi-automatically learned from data and validated by experts (Doan et al. 2003; P. Mitra, Noy, & Jaiswal 2005)).

The quality of the classifier in our setting, very likely, depends on the quality of the mappings, just as the quality of a classical classifier depends on the quality of the data (i.e., noisy data or imprecise mappings may result in very poor classifiers). In many application domains (e.g., bioinformatics), community-driven efforts are underway to develop carefully curated mappings between ontologies of interest. The cost of such efforts may be justified in some applica-
tion domains, whereas automatically derived mappings may be adequate in other domains. It should be noted that even manually derived mappings are often application, user, or context specific. Thus, users may have different views of the domain and, hence, may want to use different mappings.

The proposed algorithms for learning from OERDSs are provably exact relative to their centralized counterparts, for a family of learning classifiers for which the sufficient statistics take the form of counts of instances satisfying certain constraints on the values of the attributes.

More broadly, our research in the domain of knowledge acquisition from scientific data has led to the development of:

(a) A general theoretical framework for learning predictive models (e.g., classifiers) from large, physically distributed data sources (Caragea, Silvescu, & Honavar 2004).

(b) A theoretically sound approach to formulation and execution of statistical queries across semantically heterogeneous data sources (Caragea, Pathak, & Honavar 2004).

(c) Statistically sound approaches to learning classifiers from partially specified data resulting from data described at different levels of abstraction (Zhang, Caragea, & Honavar 2005).

(d) Tools to support collaborative development of modular ontologies (Bao, Caragea, & Honavar 2006).

(e) INDUS, a modular, extensible, open-source software toolkit for data-driven knowledge acquisition from large, distributed, autonomous, semantically heterogeneous data sources (Caragea et al. 2005a).

Related work includes several approaches to distributed learning (Park & Kargupta 2002; Kargupta et al. 1999; Srivastava et al. 1999; Domingos 1997) and information integration (Hull 1997; Davidson et al. 2001; Eckman 2003). However, to the best of our knowledge, none of these approaches combine data mining and information integration techniques into a system that can be easily used by end users to explore and extract knowledge from large, distributed, autonomous, semantically heterogeneous data sources. Our algorithms and tools have been successfully applied to data-driven knowledge acquisition tasks that arise in bioinformatics (Andorf et al. 2004; Caragea et al. 2005a; Yan, Dobbs, & Honavar 2004; Yan, Honavar, & Dobbs 2004).

Research in the area of knowledge discovery from semantically heterogeneous data is still in its infancy, posing many challenges due to the large amounts of data involved and the nature of these data. Our contributions to the general problem of knowledge acquisition from distributed, semantically heterogeneous data sources represent important steps towards solutions to problems that arise in many application domains.

Acknowledgments

This research is supported in part by grants from the National Science Foundation (0219699) and the National Institute of Health (GM 066387) to Vasant Honavar.

References


