Algorithms and Software for Collaborative Discovery from Autonomous, Semantically Heterogeneous, Distributed Information Sources

Doina Caragea, Jun Zhang, Jie Bao, Jyotishman Pathak, and Vasant Honavar

Artificial Intelligence Research Laboratory,
Center for Computational Intelligence, Learning, and Discovery,
Department of Computer Science, Iowa State University,
226 Atanasoff Hall, Ames, IA 50011
honavar@cs.iastate.edu

Abstract. Development of high throughput data acquisition technologies, together with advances in computing, and communications have resulted in an explosive growth in the number, size, and diversity of potentially useful information sources. This has resulted in unprecedented opportunities in data-driven knowledge acquisition and decision-making in a number of emerging increasingly data-rich application domains such as bioinformatics, environmental informatics, enterprise informatics, and social informatics (among others). However, the massive size, semantic heterogeneity, autonomy, and distributed nature of the data repositories present significant hurdles in acquiring useful knowledge from the available data. This paper introduces some of the algorithmic and statistical problems that arise in such a setting, describes algorithms for learning classifiers from distributed data that offer rigorous performance guarantees (relative to their centralized or batch counterparts). It also describes how this approach can be extended to work with autonomous, and hence, inevitably semantically heterogeneous data sources, by making explicit, the ontologies (attributes and relationships between attributes) associated with the data sources and reconciling the semantic differences among the data sources from a user's point of view. This allows user or context-dependent exploration of semantically heterogeneous data sources. The resulting algorithms have been implemented in INDUS - an open source software package for collaborative discovery from autonomous, semantically heterogeneous, distributed data sources.

1 Introduction

Recent development of high throughput data acquisition technologies in a number of domains (e.g., biological sciences, environmental sciences, atmospheric sciences, space sciences, commerce) together with advances in digital storage, computing, and communications technologies have resulted in the proliferation of a multitude of physically distributed data repositories created and maintained by autonomous entities (e.g., scientists, organizations). The resulting increasingly

S. Jain, H.U. Simon, and E. Tomita (Eds.): ALT 2005, LNAI 3734, pp. 13–44, 2005.

data rich domains offer unprecedented opportunities in computer assisted data-driven knowledge acquisition in a number of applications including in particular, data-driven scientific discovery in bioinformatics (e.g., characterization of protein sequence-structure-function relationships in computational molecular biology), environmental informatics, health informatics; data-driven decision making in business and commerce, monitoring and control of complex systems (e.g., load forecasting in electric power networks), and security informatics (discovery of and countermeasures against attacks on critical information and communication infrastructures). Machine learning algorithms [1, 2, 3, 4, 5, 6, 7] offer some of the most cost-effective approaches to knowledge acquisition (discovery of features, correlations, and other complex relationships and hypotheses that describe potentially interesting regularities) from large data sets. However, the applicability of current approaches to machine learning in emerging data rich applications in practice is severely limited by a number of factors:

(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 learning from multiple distributed data sources without the need to transmit large amounts of data. In other domains, the ability of autonomous organizations to share raw data may be limited due to a variety of reasons (e.g., privacy considerations). In such cases, there is a need for knowledge acquisition algorithms that can learn from statistical summaries of data (e.g., counts of instances that match certain criteria) that are made available as needed from the distributed data sources in the absence of access to raw data.
(b) Autonomously developed and operated data sources often differ in their structure and organization (relational databases, flat files, etc.) and the operations that can be performed on the data source (e.g., types of queries - relational queries, restricted subsets of relational queries, statistical queries, keyword matches; execution of user-supplied code to compute answers to queries that are not directly supported by the data source; storing results of computation at the data source for later use) and the precise mode of allowed interactions can be quite diverse. Hence, there is a need for theoretically well-founded strategies for efficiently obtaining the information needed for learning within the operational constraints imposed by the data sources.
(c) Autonomously developed data sources differ in terms of their underlying ontological commitments [8], i.e., assumptions concerning the objects that exist in the world, the properties or attributes of the objects, the possible values of attributes, and their intended meaning, as well as the granularity or level of abstraction at which objects and their properties are described. The increasing need for information sharing between organizations, individuals and scientific communities have led to significant community-wide efforts aimed at the construction of ontologies in many areas: Gene Ontology - GO (www.geneontology.org) [9] for molecular biology, Unified Medical Language System -UMLS (www.nlm.nih.gov/research/umls) for heath infor-

matics, Semantic Web for Earth and Environmental Terminology - SWEET (sweet.jpl.nasa.gov) for geospatial informatics. While explicit declaration of the ontology associated with a data repository helps standardize the semantics to an extent, because 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 repository and because data sources that are created for use in one context or application often find use in other contexts or applications, semantic differences among autonomously designed, owned, and operated data repositories are simply unavoidable. Effective use of multiple sources of data in a given context requires reconciliation of such semantic differences from the user's perspective [10, 11]. Collaborative scientific discovery applications often require users to be able to analyze data from multiple, semantically disparate data sources 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 learning (e.g., statistics) from distributed, semantically heterogeneous data based on user-specified ontologies and user-specified mappings between ontologies.

Against this background, we consider the problem of data driven knowledge acquisition from autonomous, distributed, semantically heterogeneous, data sources. The rest of this paper is organized as follows:

2 Learning from Distributed Data

2.1 Problem Formulation

Given a data set D, a hypothesis class H, and a performance criterion P, an algorithm L for learning (from centralized data D) outputs a hypothesis $h \in H$ that optimizes P. In pattern classification applications, h is a classifier (e.g., a decision tree, a support vector machine, etc.) (See Figure 1). The data D typically consists of a set of training examples. Each training example is an ordered tuple of attribute values, where one of the attributes corresponds to a class label and the remaining attributes represent inputs to the classifier. The goal of learning is to produce a hypothesis that optimizes the performance criterion e.g., minimizing classification error (on the training data) and the complexity of the hypothesis.

In a distributed setting, a data set D is distributed among the sites $1, ..., p$ containing data set fragments $D_1, ..., D_p$. Two common types of data fragmentation are: horizontal fragmentation and vertical fragmentation. In the case of horizontal fragmentation, each site contains a subset of the data tuples that make up D, i.e., $D = \cup_{i=1}^{p} D_i$. In the case of vertical fragmentation each site stores the subtuples of data tuples (corresponding to a subset of the attributes used to define data tuples in D). In this case, D can be constructed by taking the *join* of the individual data sets $D_1, ..., D_p$ (assuming a unique identifier for each

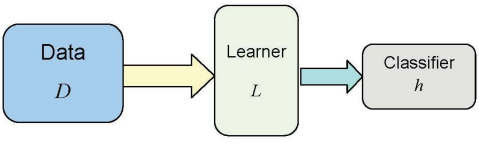


Fig. 1. Learning from centralized data

data tuple is stored with the corresponding subtuples). More generally, the data may be fragmented into a set of relations (as in the case of tables of a relational database, but distributed across multiple sites) i.e., $D = \bigotimes_{i=1}^{p} D_i$ (where $\bigotimes$ denotes the *join* operation). If a data set D is distributed among the sites $1, ..., p$ containing data set fragments $D_1, ..., D_p$, we assume that the individual data sets $D_1, ..., D_p$ collectively contain (in principle) all the information needed to construct the dataset D. More generally, D may be fragmented across multiple relations [12, 13, 14, 15, 16].

The distributed setting typically imposes a set of constraints Z on the learner that are absent in the centralized setting. For example, the constraints Z may prohibit the transfer of raw data from each of the sites to a central location while allowing the learner to obtain certain types of statistics from the individual sites (e.g., counts of instances that have specified values for some subset of attributes). In some applications of data mining (e.g., knowledge discovery from clinical records), Z might include constraints designed to preserve privacy.

The problem of learning from distributed data can be stated as follows: Given the fragments $D_1, ..., D_p$ of a data set D distributed across the sites $1, ..., p$, a set of constraints Z, a hypothesis class H, and a performance criterion P, the task of the learner L_d is to output a hypothesis that optimizes P using only operations allowed by Z. Clearly, the problem of learning from a centralized data set D is a special case of learning from distributed data where $p = 1$ and $Z = \phi$. Having defined the problem of learning from distributed data, we proceed to define some criteria that can be used to evaluate the quality of the hypothesis produced by an algorithm L_d for learning from distributed data relative to its centralized counterpart. We say that an algorithm L_d for learning from distributed data sets $D_1, ..., D_p$ is exact relative to its centralized counterpart L if the hypothesis produced by L_d is identical to that obtained by L from the data set D obtained by appropriately combining the data sets $D_1, ..., D_p$.

Example: Let L_d be an algorithm for learning a Support Vector Machine (SVM) classifier [Cortes and Vapnik, 1995] $h_d : R^N \rightarrow \{-1, 1\}$ under a set of specified constraints Z from horizontally fragmented distributed data $D_1, ..., D_p$, where each $D_i \subset D \subset R^N \times \{-1, 1\}$. Let L be a centralized algorithm for learning an SVM classifier $h : R^N \rightarrow \{-1, 1\}$ from data set $D \subset R^N \times \{-1, 1\}$. Suppose further that $D = \cup_{i=1}^{p} D_i$ Then we say that L_d is exact with respect to L if and only if $\forall X \in R^N, h(X) = h_d(X)$.

Proof of exactness of an algorithm for learning from distributed data relative to its centralized counterpart ensures that a large collection of existing theoretical (e.g., sample complexity, error bounds) as well as empirical results

obtained in the centralized setting apply in the distributed setting. We can define exactness of learning from distributed data with respect to other criteria of interest (e.g., expected accuracy of the learned hypothesis). More generally, it might be useful to consider algorithms for learning from distributed data that are provably approximate relative to their centralized counterparts. However, in the discussion that follows, we focus on algorithms for learning from distributed data that are provably exact with respect to their centralized counterparts in the sense defined above.

2.2 A General Framework for Designing Algorithms for Learning from Distributed Data

Our general strategy for designing an algorithm for learning from distributed data that is provably exact with respect to its centralized counterpart (in the sense defined above) follows from the observation that most of the learning algorithms use only certain statistics computed from the data D in the process of generating the hypotheses that they output. (Recall that a statistic is simply a function of the data. Examples of statistics include mean value of an attribute, counts of instances that have specified values for some subset of attributes, the most frequent value of an attribute, etc.) This yields a natural decomposition of a learning algorithm into two components:

(a) an information extraction component that formulates and sends a statistical query to a data source and
(b) a hypothesis generation component that uses the resulting statistic to modify a partially constructed hypothesis (and further invokes the information extraction component as needed).

A statistic $s(D)$ is called a sufficient statistic for a parameter θ if $s(D)$, loosely speaking, provides all the information needed for estimating the parameter from data D. Thus, sample mean is a sufficient statistic for the mean of a Gaussian distribution. A sufficient statistic s for a parameter θ is called a minimal sufficient statistic if for every sufficient statistic s_θ for θ, there exists a function $g_{s_\theta}(s_\theta(D)) = s(D)$ [17, 18].

We have, inspired by theoretical work on PAC learning from statistical queries [19], generalized this notion of a sufficient statistic for a parameter θ into a sufficient statistic $s_{L,h}(D)$ for learning a hypothesis h using a learning algorithm L applied to a data set D [20].

Trivially, the data D is a sufficient statistic for learning h using L. However, we are typically interested in statistics that are minimal or at the very least, substantially smaller in size (in terms of the number of bits needed for encoding) than the data set D. In some simple cases, it is possible to extract a sufficient statistic $s_{L,h}(D)$ for constructing a hypothesis h in one step (e.g., by querying the data source for a set of conditional probability estimates when L is the standard algorithm for learning a Naive Bayes classifier). In such a case, we say that $s_{L,h}(D)$ is a sufficient statistic for learning h using the learning algorithm L if there exists an algorithm that accepts $s_{L,h}(D)$ as input and outputs $h = L(D)$.

In a more general setting, h is constructed by L by interleaving information extraction (statistical query) and hypothesis generation operations. For example, a decision tree learning algorithm would start with an empty initial hypothesis h_0, obtain the sufficient statistics (expected information concerning the class membership of an instance associated with each of the attributes) for the root of the decision tree (a partial hypothesis h_1), and recursively generate queries for additional statistics needed to iteratively refine h_1 to obtain a succession of partial hypotheses h_1, h_2 culminating in h (See Figure 2).

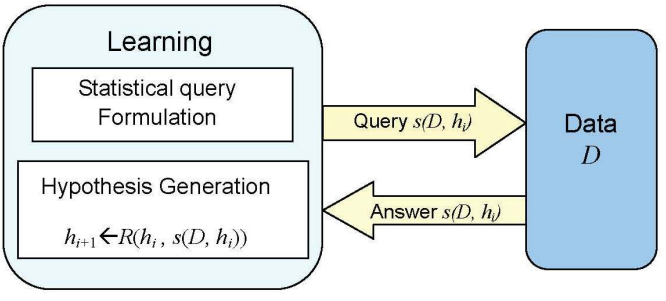


Fig. 2. Learning = Statistical Query Answering + Hypothesis Generation

We say that $s(D, h_i)$ is a sufficient statistic for the *refinement* of a hypothesis h_i into h_{i+1} (denoted by $s_{h_i \to h_{i+1}}$) if there exists an algorithm R which accepts h_i and $s(D, h_i)$ as inputs and outputs h_{i+1} . We say that $s_h(D, h_1, ..., h_m)$ is a sufficient statistic for the *composition* of the hypotheses $(h_1 ... h_m)$ into h (denoted by $s_{(h_1,...,h_m) \to h}$) if there exists an algorithm C which accepts as inputs $h_1 ... h_m$ and $s_h(D, h_1, ..., h_m)$ and outputs the hypothesis h. We say that $s_{h_i \to h_{i+k}}$ (where $i \geq 0$ and $k > 0$ are positive integers) is a sufficient statistic for iteratively refining a hypothesis h_i into h_{i+k} if h_{i+k} can be obtained through a sequence of refinements starting with h_i . We say that $s_{(h_1,...,h_m) \to h}$ is a sufficient statistic for obtaining hypothesis h starting with hypotheses $h_1, ..., h_m$ if h can be obtained from $h_1, ..., h_m$ through some sequence of applications of composition and refinement operations. Assuming that the relevant sufficient statistics (and the procedures for computing them) can be defined, the application of a learning algorithm L to a data set D can be reduced to the computation of $s_{(h_0,...,h_m) \to h}$) through some sequence of applications of hypothesis refinement and composition operations starting with the hypothesis h (See Figure 3). In this model, the only interaction of the learner with the repository of data D is through queries for the relevant statistics. Information extraction from distributed data entails decomposing each statistical query q posed by the information extraction component of the learner into sub queries $q_1, ..., q_n$ that can be answered by the individual data sources $D_1, ..., D_p$ respectively, and a procedure for combining the answers to the sub queries into an answer for the original query q. (See Figure 3).

It is important to note that the general strategy for learning classifiers from distributed data is applicable to a broad class of algorithms for learning classi-

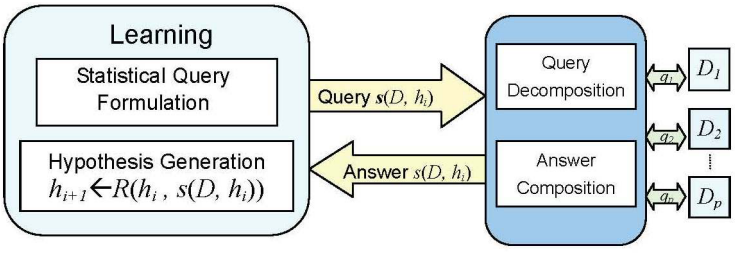


Fig. 3. Learning from Distributed Data = Statistical Query Answering + Hypothesis generation

fiers from data[20]. This follows from the fact that the output h of any learning algorithm is in fact a function of the data D, and hence by definition, a statistic. Consequently, we can devise a strategy for computing h from the data D through some combination of refinement and composition operations starting with an initial hypothesis (or an initial set of hypotheses). When the learner's access to data sources is subject to constraints Z, the resulting plan for information extraction has to be executable without violating the constraints Z. The exactness of the algorithm L_d for learning from distributed data relative to its centralized counterpart, which requires access to the complete data set D follows from the correctness (soundness) of the query decomposition and answer composition procedure. The separation of concerns between hypothesis construction and extraction of sufficient statistics from data makes it possible to explore the use of sophisticated techniques for query optimization that yield optimal plans for gathering sufficient statistics from distributed data sources under a specified set of constraints that describe the query capabilities of the data sources, operations permitted by the data sources (e.g., execution of user supplied procedures), and available computation, bandwidth, and memory resources.

2.3 Representative Algorithms for Learning Classifiers from Distributed Data

We have applied the general framework described above for construction of algorithms for learning classifiers from distributed data to design provably exact algorithms for learning Naive Bayes, Nearest Neighbor, and Decision Tree classifiers from distributed data under horizontal as well as vertical data fragmentation [21], and Support Vector Machine (SVM) Classifiers under horizontal data fragmentation [22, 23]. We briefly summarize our results on learning decision tree classifiers and SVM classifiers from distributed data We have obtained similar results for algorithms for learning Naive Bayes, Neural Network, and Bayesian Network classifiers [24].

Algorithms for Learning Decision Tree Classifiers from Distributed Data. Algorithms that construct decision tree classifiers [25, 26] search in a greedy fashion for attributes that yield the maximum amount of information for determining the class membership of instances in a training set D of labeled

instances. The information gain associated with an attribute under consideration at a particular node can be expressed in terms of the relative frequencies of instances that satisfy certain constraints on attribute values (determined by the path from the root to each of the nodes resulting from the split) for each possible class [21, 27, 28]. We have devised provably sound strategies for gathering the necessary statistics from distributed data sets, thereby obtaining distributed decision tree learning algorithms that are provably exact relative to their centralized counterparts [21]. This approach to learning decision trees from distributed data provides an effective way to learn classifiers in scenarios in which the distributed data sources provide only statistical summaries of the data and the set of unique keys on demand but prohibit access to data instances. Even when it is possible to access the raw data, the proposed algorithm compares favorably with the centralized counterpart which needs access to the entire data set whenever the communication cost incurred by the former is lower than the cost of collecting the entire data set in a central location. Let $|D|$ be the total number of examples in the distributed data set; $|A|$, the number of attributes; V the maximum number of possible values per attribute; p the number of sites across which the data set D is distributed; M the number of classes; and $size(T)$ the number of nodes in the decision tree. Our analysis [20] has shown that in the case of horizontally fragmented data, the distributed algorithm has an advantage when $MVp\ size(T) < |D|$. In the case of vertically fragmented data, the corresponding conditions are given by $size(T) < |A|$. Our experiments have shown that these conditions are often met in the case of large, high-dimensional data sets that are encountered in several applications (e.g., construction of decision trees for classification of protein sequences into functional families) [29, 30] in computational biology.

Learning Support Vector Machine Classifiers from Distributed Data. Support Vector Machine (SVM) algorithm [31, 32] constructs a binary classifier that corresponds to a separating hyperplane that maximizes the margin of separation in R^N between instances belonging two classes. Because the weight vector that defines the maximal margin hyperplane can be expressed as a weighted sum of a subset of training instances (called support vectors), the support vectors and the associated weights also constitute a sufficient statistic for SVM. In a distributed setting under horizontal fragmentation of data, it is possible to compute the maximal margin separating hyperplane by making several passes through the distributed data sets (without having to gather all of the data in a centralized place), and updating the hyperplane on each pass so as to maximize the margin of separation. We have shown (based on convergence results for SVM algorithms proved by [33]) that this strategy yields a provably exact algorithm for learning an SVM classifier from distributed data under horizontal fragmentation [22, 23].

2.4 Related Work on Learning Classifiers from Distributed Data

Srivastava et al. [34] propose methods for distributing a large centralized data set to multiple processors to exploit parallel processing to speed up learning.

Grossman and Guo [35], and Provost and Kolluri [36] survey several methods that exploit parallel processing for scaling up data mining algorithms to work with large data sets. In contrast, the focus of the our work is on learning classifiers from a set of autonomous distributed data sources. The autonomous nature of the data sources implies that the learner has little control over the manner in which the data are distributed among the different sources.

Distributed data mining has received considerable attention in the literature [37]. Domingos [38] and Prodromidis et al. [39] propose an *ensemble of classifiers* approach to learning from horizontally fragmented distributed data which essentially involves learning separate classifiers from each data set and combining them typically using a weighted voting scheme. This requires gathering a subset of data from each of the data sources at a central site to determine the weights to be assigned to the individual hypotheses (or shipping the ensemble of classifiers and associated weights to the individual data sources where they can be executed on local data to set the weights). In contrast, our approach is applicable even in scenarios which preclude transmission of data or execution of user-supplied code at the individual data sources but allow transmission of minimal sufficient statistics needed by the learning algorithm. A second potential drawback of the ensemble of classifiers approach to learning from distributed data is that the resulting ensemble of classifiers is typically much harder to comprehend than a single classifier. A third important limitation of the ensemble classifier approach to learning from distributed data is the lack of strong guarantees concerning accuracy of the resulting hypothesis relative to the hypothesis obtained in the centralized setting.

Bhatnagar and Srinivasan [40] propose an algorithm for learning decision tree classifiers from vertically fragmented distributed data. Kargupta et al. [41] describe an algorithm for learning decision trees from vertically fragmented distributed data using a technique proposed by Mansour [42] for approximating a decision tree using *Fourier coefficients* corresponding to attribute combinations whose size is at most logarithmic in the number of nodes in the tree. At each data source, the learner estimates the Fourier coefficients from the local data, and transmits them to a central site. These estimates are combined to obtain a set of Fourier coefficients for the decision tree (a process which requires a subset of the data from each source to be transmitted to the central site). A given set of Fourier coefficients can correspond to multiple decision trees. At present, such approaches offer no guarantees concerning the performance of the hypothesis obtained in the distributed setting relative to that obtained in the centralized setting.

Unlike the papers summarized above, our approach summarized in Section 2.2 [20] offers a general framework for the design of algorithms for learning from distributed data that is provably exact with respect to its centralized counterpart. Central to our approach is a clear separation of concerns between hypothesis construction and extraction of sufficient statistics from data, making it possible to explore the use of sophisticated techniques for query optimization that yield optimal plans for gathering sufficient statistics from distributed data

sources under specified set of constraints Z that describe the query capabilities and operations permitted by the data sources (e.g., execution of user supplied procedures). Our approach also lends itself to adaptation to learning from semantically heterogeneous data sources (see below). Provided the needed mappings between ontologies can be specified, our approach to learning from distributed data can be extended to yield a sound approach to learning from heterogeneous distributed data encountered in practical applications (see Section 3).

3 Information Integration from Semantically Heterogeneous Distributed Data

3.1 Semantic Data Integration Problem

In order to extend our approach (summarized in Section 2.2) to learning from distributed data (which assumes a common ontology that is shared by all of the data sources) into effective algorithms for learning classifiers from *semantically heterogeneous* distributed data sources, techniques need to be developed for answering the statistical queries posed by the learner in terms of the learner's ontology O from the heterogeneous data sources (where each data source Di has an associated ontology O_i). Thus, we have to solve a variant of the problem of integrated access to distributed data repositories - the data integration problem [43] in order to be able to use machine learning approaches to acquire knowledge from semantically heterogeneous data. This problem is best illustrated by an example: Consider two academic departments that independently collect information about their *Students*. Suppose a data set D_1 collected by the first department is described by the attributes *ID*, *Student Level*, *Monthly Income* and *Internship* and it is stored into a table as the one corresponding to D_1 in Table 1. Suppose a data set D_2 collected by the second department is described by the attributes *Student ID*, *Student Program*, *Hourly Income* and *Intern* and it is stored into a table as the one corresponding to D_2 in Table 1.

Table 1. Student data collected by two departments and a university statistician

	ID	*Student Level*	*Monthly Income*	*Internship*
D_1	34	M.S.	1530	yes
	49	1st Year	600	no
	23	Ph.D.	1800	no

	SID	*Student Program*	*Hourly Income*	*Intern*
D_2	1	Master	14	yes
	2	Doctoral	17	no
	3	Undergraduate	8	yes

	SSN	*Student Status*	*Yearly Income*	*Intern*
D_U	475	Master	16000	?
	287	Ph.D.	18000	?
	530	Undergrad	7000	?

Consider a user, e.g., a university statistician, who wants to construct a predictive model based on data from two departments of interest from his or her own perspective, where the representative attributes are *Student SSN*, *Student Status*, *Yearly Income* and *Industry Experience*. For example, the statistician may want to construct a model that can be used to infer whether a typical student (represented as in the entry corresponding to D_U in Table 1) drawn from the same population from which the two departments receive their students is likely to have completed an internship. This requires the ability to perform queries over the two data sources associated with the departments of interest from the user's perspective (e.g., *fraction of doctoral students who completed an internship*). However, because the two data sources differ in terms of semantics from the user's perspective the user must recognize the semantic correspondences between the attributes *ID* in the first data source, *Student ID* in the second data source and *Student SSN* in the user data; the attributes *Student Level*, *Student Program* and *Student Status*, etc. From our perspective, a data integration system should: allow users to specify what information is needed instead of how it can be obtained; allow each user to impose his or her own points of view (ontological commitments) on the data sources and post queries specified using terms in that ontology; hide the complexity of communication and interaction with heterogeneous distributed data sources; automatically transform user queries into queries that are understood by the respective data sources; map the results obtained into the form expected by the user and store them for future analysis; allow incorporation of new data sources as needed; and support sharing of ontologies (hence ontological commitments) and among users as needed [10].

3.2 INDUS: An Ontology Based Federated Query Centric Data Integration System

Our recent work has led to the development of a *federated, query-centric* approach to information integration from heterogeneous, distributed information sources which has been implemented in the data integration component of INDUS (Intelligent Data Understanding System) prototype [10, 11, 44] (See Figure 4).

The choice of the federated (as opposed to data warehouse) and query centric (as opposed to source centric) approach to information integration was motivated by characteristics of a class of scientific applications of data-driven knowledge acquisition. A detailed discussion of the design rationale of INDUS can be found in [10, 20, 44]. In brief, a federated approach lends itself much better to settings where it is desirable to postpone specification of user ontology O and the mapping $M(O, O_1, \cdots, O_p) = \{M(O, O_1), \cdots M(O, O_p)\}$ between O and data source specific ontologies $O_1, \cdots, O_p$ until when the user is ready to use the system. The choice of a query centric approach in INDUS enables users the desired flexibility in integrating data from multiple autonomous sources in ways that match their own context or application specific ontological commitments whereas in a source centric approach, the semantics of the data (what the data from a source should mean to a user) are determined by the source. INDUS enables a scientist

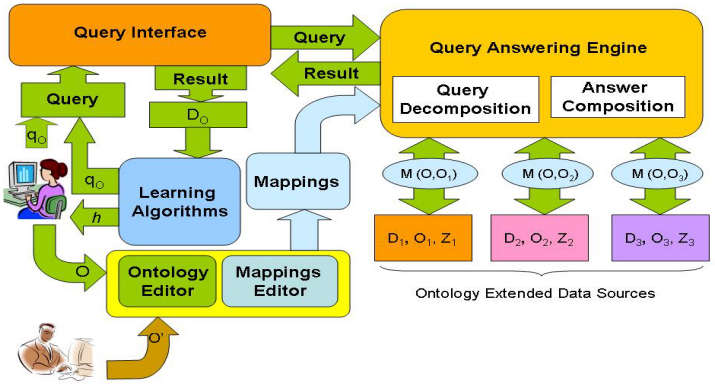
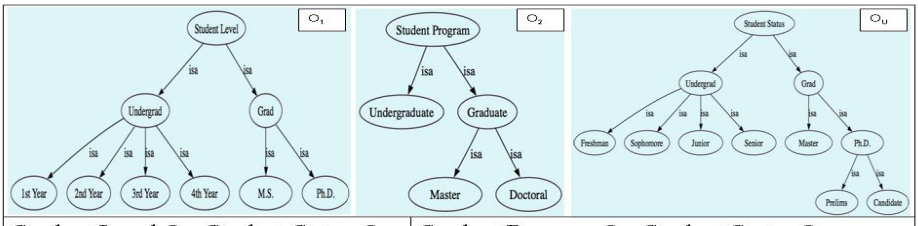


Fig. 4. INDUS (Intelligent Data Understanding System) for Information Integration from Heterogeneous, Distributed, Autonomous Information Sources. D_1, D_2, D_3 are data sources with associated ontologies O_1, O_2, O_3 and O is a user ontology. Queries posed by the user are answered by a query answering engine in accordance with the mappings between user ontology and the data source ontologies, specified using a user-friendly editor.

to view a collection of physically distributed, autonomous, heterogeneous data sources (regardless of their location, internal structure, and query interfaces) as *though* they were relational databases, (i.e. a collection of inter-related tables. Each data source in INDUS has associated with it, a data source description which includes the ontology of the data source and a description of the query capabilities of the data source (i.e., the schema of the data source). INDUS makes explicit the (sometimes implicit) ontologies associated with data sources. This allows the specification of semantic correspondences between data sources [11] which can be expressed in ontology-extended relational algebra (independently developed by [45]).

We assume that each data source has associated with it, an ontology that includes hierarchies corresponding to attribute value taxonomies (AVT) (See Figure 5). We specify the correspondence between semantically similar attributes, by mapping the domain of the type of one attribute to the domain of the type of the semantically similar attribute (e.g., *Hourly Income* to *Yearly Income* or *Student Level* to *Student Status*) [11]. Explicit specification of mappings between AVTs in the user ontology O_U and data source ontologies O_1 and O_2 allows the user to view data D_1 and D_2 from his or her own perspective. Such mappings can be used to answer user queries that are expressed in terms of O_U from the data sources D_1 and D_2. Let $< D_1, O_1, S_1 >, \cdots, < D_p, O_p, S_p >$ be an ordered set of p ontology-extended data sources and U a user that poses queries against the heterogeneous data sources $D_1, \cdots, D_p$. A user perspective P_U is given by a user ontology O_U and a set of *semantic correspondences* or *interoperation constraints* IC that define relationships between terms in $O_1, \cdots, O_p$, respectively, and the terms in O_U. The semantic correspondences take one of the two forms: $x \leq y$



Student Level:O_1≡Student Status:O_U
Undergrad:O_1≡Undergrad:O_U
1st Year:O_1≡Freshman:O_U
2nd Year:O_1≡Sophomore:O_U
3rd Year:O_1≡Junior:O_U
4th Year:O_1≡Senior:O_U
Grad:O_1≡Grad:O_U
M.S:O_1≡Master:O_U
Ph.D.:O_1≡Ph.D.:O_U

Student Program:O_2≡Student Status:O_U
Undergraduate:O_2≡Undergrad:O_U
Graduate:O_2≡Grad:O_U
Master:O_2≡Master:O_U
Doctoral:O_2≡Ph.D.:O_U

Fig. 5. Attribute value taxonomies (ontologies) O_1 and O_2 associated with the attributes *Student Level*, *Student Program* in two data sources of interest. O_U is the ontology for *Student Status* from the user's perspective. An example of user-specified semantic correspondences between the user ontology O_U and data source ontologies O_1 and O_2 respectively is also shown.

(x is semantically subsumed by y), $x \geq y$ (x semantically subsumes y), $x \equiv y$ (x is semantically equivalent to y), $x \neq y$ (x is semantically incompatible with y), $x \approx y$ (x is semantically compatible with y) (inspired by bridge rules introduced by Bouquet et al. [46]). See 5 for an illustration of user-defined semantic correspondences between data sources O_1 and O_2, respectively, and O_U.

Let $O_1, \cdots, O_p$ (respectively) be the ontologies associated with the data sources $D_1, \cdots, D_p$. Let $P_U = (O_U, IC)$ a user perspective with respect to these ontologies. We say that the ontologies $O_1, \cdots O_p$, are integrable according to the user ontology O_U in the presence of semantic correspondences IC if there exist p partial injective mappings $M(O_U, O_1), \cdots, M(O_U, O_p)$ from $O_1, \cdots, O_p$, respectively, to O_U. Examples of such mappings include functions for converting monthly income and hourly income (respectively) from the ontologies associated with data sources D_1 and D_2 (see Figure 5) into yearly income in terms of user ontology O_U; or for mapping instances corresponding to *1st year* students from data source D_1 into instances described as *Freshman* from the user perspective. We have completed the implementation of a working prototype of the INDUS system to enable users with some familiarity with the relevant data sources to rapidly and flexibly assemble data sets from multiple data sources and to query these data sets. This can be done by specifying a user ontology, simple semantic mappings between data source specific ontologies and the user ontolgy and queries - all without having to write any code. The current implementation of INDUS which has been released under Gnu public license (http://www.cild.iastate.edu/software/indus.html) includes support for:

(a) Import and reuse of selected fragments of existing ontologies and editing of ontologies [47].

(b) Specification of semantic correspondences between a user ontology O_U and data source ontologies [11]. Semantic correspondences between ontologies can be defined at two levels: schema level (between attributes that define data source schemas) and attribute level (between values of attributes). Consistency of semantic correspondences is verified by reasoning about subsumption and equivalence relationships [48]
(c) Registration of a new data source using a data-source editor for defining the schema of the data source (the names of the attributes and their corresponding ontological types), location, type of the data source and access procedures that can be used to interact with a data source.
(d) Specification and execution of queries across multiple semantically heterogeneous, distributed data sources with different interfaces, functionalities and access restrictions. Each user may choose relevant data sources from a list of data sources that have been previously registered with the system and specify a user ontology (by selecting an ontology from a list of available ontologies or by invoking the ontology editor and defining a new ontology). The user can select mappings between data source ontologies and user ontology from the available set of existent mappings (or invoke the mapings editor to define a new set of mappings). The data needed for answering a query is specified by selecting (and possibly restricting) attributes from the user ontology, through a user-friendly interface. Queries posed by the user are sent to a query-answering engine (QAE) that automatically decomposes the user query expressed in terms of the user ontology into queries that can be answered by the individual data sources. QAE combines (after applying the necessary mappings) to generate the answer for the user query. The soundness of the data integration process (relative to a set of user-specified mappings between ontologies) follows from the soundness of the query decomposition procedure, and the correctness of the behavior of the query answering engines associated with the individual data sources, and the answer composition procedure [11].
(e) Storage and further manipulation of results of queries. The results returned by a user query can be temporarily stored in a local relational database. This in effect, represents a materialized relational view (modulo the mappings between user and data source specific ontologies) across distributed, heterogeneous (and not necessarily relational) data repositories. The current design of INDUS supports further analysis (e.g., by applying machine learning algorithms) on the retrieved data.

In summary, INDUS offers the functionality necessary to flexibly integrate information from multiple heterogeneous data sources and structure the results according to a user-supplied ontology. INDUS has been used to assemble several data sets used in the exploration of protein sequence-structure-function relationships [44].

3.3 Related Work on Data Integration

Hull [49], Davidson et al. [50] and Eckman [51] survey alternative approaches to data integration. A wide range of approaches to data integration have been

considered including multi-database systems [52, 53, 54], mediator based approaches [55, 56, 57, 58, 59, 60, 61, 62]. Several data integration projects have focused specifically on integration of biological data [63, 64, 65, 66, 67]. Tomasic et al. [68] proposed an approach to scaling up access to heterogeneous data sources. Haas et al. [69] investigated optimization of queries across heterogeneous data sources. Space does not permit a detailed survey of the extensive literature on data integration. Rodriguez-Martinez and Roussoloulos [70] proposed a code shipping approach to design of an extensible middleware system for distributed data sources. Lambrecht et al. [71] proposed a planning framework for gathering information from distributed sources. These efforts addressed, and to varying degrees, solved the following problems in data integration: design of query languages and rules for decomposing queries into sub queries and composing the answers to sub queries into answers to the initial query. Maluf and Wiederhold [72] proposed an ontology algebra for merging of ontologies. Our group developed an approach to specifying semantic correspondences between ontologies and for querying semantically heterogeneous data using ontologies and inter-ontology mappings [10]. This approach is similar to the ontology-extended relational algebra developed by Bonatti et al. [45]. The design of INDUS [10, 11, 44] was necessitated by the lack of publicly available data integration platforms that could be used as a basis for learning classifiers from semantically heterogeneous distributed data. INDUS draws on much of the existing literature on data integration and hence shares some of the features of existing data integration platforms. But it also includes some relatively novel features (See Section 3.2).

4 Knowledge Aquisition from Semantically Heterogeneous Distributed Data

The stage is now set for developing sound approaches to learning from semantically heterogeneous, distributed data (See Figure 6). While it is possible to retrieve the *data* necessary for learning from a set of heterogeneous data sources using INDUS, store the retrieved data in a local database, and then apply standard (centralized) learning algorithms, such approach is not feasible when the amounts of data involved are large, and bandwidth and memory are limited, or when the query capabilities of the data sources are limited to answering a certain class of statistical queries (e.g., counts of instances that satisfy certain constraints on the values of their attributes). Hence, the development of sound approaches to answering statistical queries from semantically heterogeneous data sources a variety of constraints and assumptions motivated by application scenarios encountered in practice is a key element of our research plan.

4.1 Partially Specified Data

Our approach to design of algorithms for learning classifiers from semantically heterogeneous distributed data is a natural extension of our approach to learning from distributed data discussed in Section 2 (See Figure 3) which assumed a common ontology that is shared by all of the data sources. We propose to extend this

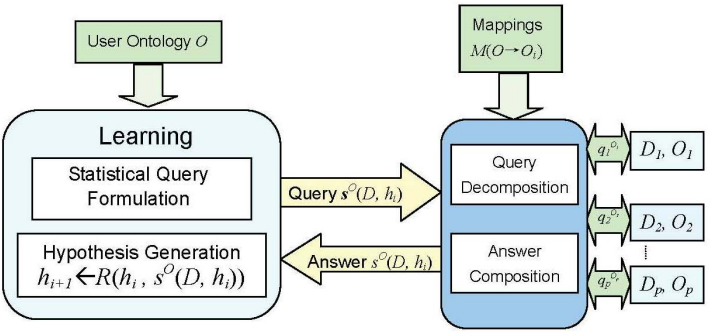


Fig. 6. General Framework for learning classifiers from semantically heterogeneous distributed data. The learner generates statistical queries expressed in terms of user ontology). These queries have to be mapped into queries expressed in terms of data source specific ontologies that can be executed directly on the data sources and the results combined and mapped into the answer to the query posed by the learner.

framework to work with semantically heterogeneous distributed data sources by developing techniques for answering the statistical queries posed by the learner in terms of the learner's ontology O using the heterogeneous data sources (where each data source D_i has an associated ontology O_i) (See Figure 6).

Before we can discuss approaches for answering statistical queries from semantically heterogeneous data, it is useful to explore what it means to answer a statistical query in a setting in which autonomous data sources differ in terms of the levels of abstraction at which data are described. We illustrate some of the issues that have to be addressed using an example: Consider the data source ontologies O_1 and O_2 and the user ontology O_U shown in Figure 5. The attribute *Student Status* in data source D_1 is specified in greater detail (lower level of abstraction) than in D_2. That is, data source D_1 carries information about the precise categorization of *Undergrad* students into 1^{st} year, 2^{nd} year, 3^{rd} year, and 4^{th} year students, whereas data source D_2 makes no such distinctions among *Undergraduate* students. Now suppose that D_1 contains 5, 10, 15, 10 instances (respectively) of 1^{st} year, 2^{nd} year, 3^{rd} year, and 4^{th} year (undergrad) students and 20 instances of *Grad* students. Suppose D_2 contains 20 instances of *Undergraduate* students, 40 instances of *Graduate* students respectively.

Suppose a statistical query q^{O_U} is posed by the user against the two data sources D_1 and D_2 based on ontology O_U: What fraction of students are *Undergrads*? The answer to this query can be computed in a straightforward fashion as the ratio of number of *Undergrad students* ((5+10+15+10)+20=60) divided by the total number of students whose *Student Status* is recorded in the two data sources (60+20+40=120) yielding an answer of 0.5.

Now consider a different statistical query r^{O_U}: What fraction of the students in the two data sources are *sophomores*? The answer to this query is not as straightforward as the answer to the previous query q^{O_U}. This is due to the fact that the *Student Status* of student records in data source D_2 are only *partially*

specified [73, 74] with respect to the ontology O. Consequently, we can never know the precise fraction of students that are *Sophomores* based on the information available in the two data sources. However, if it is reasonable to assume that the data contained in both D_1 and D_2 are drawn from the same *universe* (i.e., can be modeled by the same underlying distribution), we can *estimate* the fraction of students that are *Sophomores* in the data source D_2 based on the fraction of *Undergrad* students that are *Sophomores* in the data source D_1 (i.e., 10 out of 40) and use the result to answer the query r^{O_U}. Under the assumption that the population of students in D_1 and D_2 can be modeled by the same distribution, the *estimated* number of *Sophomore* students in D_2 is given by $(\frac{10}{40})(20) = 5$. Hence, the *estimated* number of *Sophomore* students in D_1 and $D_2 = 10+5 = 15$. Thus, the answer to the query r^{O_U} is $\frac{15}{120} = 0.125$. Note that the answer to query q^{O_U} is completely determined by the data source ontologies O_1, O_2, the user ontology O_U and the mappings shown in Figure 5. However, answer to the query r^{O_U} is only partially determined by the ontologies and the mappings shown in Figure 5. In such cases, answering statistical queries from semantically heterogeneous data sources requires the user to supply not only the mapping between the ontology and the ontologies associated with the data sources but also additional assumptions of a statistical nature (e.g., that data in D_1 and D_2 can be modeled by the same underlying distribution) and the validity of the answer returned depends on the validity of the assumptions and the soundness of the procedure that computes the answer based on the supplied assumptions.

Hence, the development of algorithms for learning classifiers from semantically heterogeneous data requires addressing the problem of learning classifiers from partially specified data. Specifically, this entails development provably sound methods based extensions to our current formulations of ontology-based query decomposition and answer composition methods in INDUS [11] for answering statistical queries from semantically heterogeneous data sources under alternative statistical assumptions.

4.2 The Problem of Learning Classifiers from Partially Specified Data

Let us start by considering a *partially specified* centralized data set D with an associated ontology O. Let $\{A_1, A_2, ..., A_n\}$ be an ordered set of nominal attributes, and let $dom(A_i)$ denote the set of values (the domain) of attribute A_i. An attribute value taxonomy T_i for attribute A_i is a tree structured concept hierarchy in the form of a partially order set $(dom(A_i), \leq)$, where $dom(A_i)$ is a finite set that enumerates all attribute values in A_i, and $\leq$ is the partial order that specifies *isa* relationships among attribute values in $dom(A_i)$ (see any of the ontologies in Figure 5). Collectively, $O = \{T_1, T_2, ..., T_n\}$ represents the ordered set of attribute value taxonomies associated with attributes $\{A_1, A_2, ..., A_n\}$ (see Figure 7).

Let $Nodes(T_i)$ represent the set of all values in T_i, and $Root(T_i)$ stand for the root of T_i. The set of leaves of the tree, $Leaves(T_i)$, corresponds to the set of *primitive values* of attribute A_i (e.g., Freshman, Sophomore, etc. in the hierarchy corresponding to the attribute *Student Status* in Figure 5). The internal

nodes of the tree (i.e., $Nodes(T_i) - Leaves(T_i)$) correspond to *abstract values* of attribute A_i (e.g., Undergrad, Grad, Ph.D. in in the hierarchy corresponding to the attribute *Student Status* Figure 5). Each arc of the tree corresponds to a *isa* relationship over attribute values in the AVT. Thus, an AVT defines an abstraction hierarchy over values of an attribute.

The set of abstract values at any given level in the tree T_i form a partition of the set of values at the next level (and hence, the set of primitive values of A_i). For example, in Figure 5, the nodes at level 1, i.e., *Undergrad*, *Grad*, define a partition of attribute values that correspond to nodes at level 2 (and hence, a partition of all primitive values of the *Student Status* attribute). After Haussler [75], we define a cut γ_i of an AVT T_i as a subset of nodes in T_i satisfying the following two properties: (1) For any leaf $l \in Leaves(T_i)$, either $l \in \gamma_i$ or l is a descendent of a node $n \in \gamma_i$; and (2) For any two nodes $f, g \in \gamma_i$, f is neither a descendent nor an ancestor of g. Cuts through AVT T_i correspond to a partition of $Leaves(T_i)$. Thus, the *cut* corresponding to *Undergrad, Master, Ph.D.* defines a partition over the primitive values of the *Student Status* attribute.

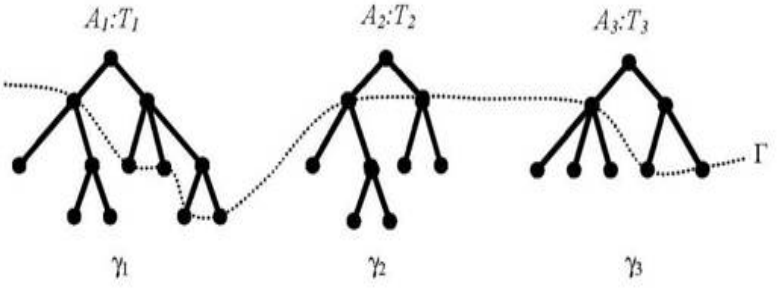


Fig. 7. Global cut through a set of attribute value taxonomies

The original instance space I in the absence of AVT is an instance space defined over the domains of all attributes. Let $\Gamma = \{\gamma_1, ..., \gamma_n\}$ be a global cut through T, where γ_i stands for a cut through T_i (see Figure 7). The cut Γ defines an *abstract instance space* I_Γ. A set of AVT $O = \{T_1, T_2, ..., T_n\}$ associated with a set of attributes $A = \{A_1, A_2, ..., A_n\}$ induces an instance space $I_O = \cup_\Gamma I_\Gamma$ (the union of instance spaces induced by all the cuts through the set of AVT O). We say that an instance $x \in I_O$ is *partially specified* if one or more of its attribute values are not primitive. A *partially specified data set* D_O (relative to a set O of AVT) is a collection of instances drawn from I_O where each instance is labeled with the appropriate class label from $C = \{c_1, c_2, ..., c_k\}$, a finite set of mutually disjoint classes. Thus, $D_O \subset I_O \times C$.

The problem of learning classifiers from AVT and partially specified data can be formulated as follows: Given a user-supplied set of AVT O and a data set D_O of (possibly) partially specified labeled instances, construct a classifier $h_O : I_O \rightarrow C$ for assigning appropriate class labels to each instance in the instance space I_O.

4.3 Learning from Partially Specified Semantically Heterogeneous Data

Suppose that a data set D is distributed over the data sources $D_1, \cdots, D_p$, where each data source D_i contains only a horizontal fragment (subset of data tuples) of the data D. Each distributed data set D_i is described by the set of attributes $\{A_1^i, \cdots, A_n^i\}$ and their corresponding AVT $O_i = \{T_1^i, \cdots, T_n^i\}$. Let $\{A_1^U, \cdots, A_n^U\}$ be the set of attributes that describe the data D in a user view and let $O_U = T_1^U, ..., T_n^U$ be a user-supplied collection of taxonomies over the set of attributes $A_1^U, ..., A_n^U$. Let $\Psi = \{\varphi_1, \varphi_2, \cdots, \varphi_p\}$ be a collection of user-defined mappings from data source taxonomies O_i to user taxonomies O_U, respectively. A global cut Γ^U in the user's collection of taxonomies $O_U = \{T_1^U, \cdots, T_n^U\}$ determines cuts $\{\Gamma_1, \Gamma_2, \cdots, \Gamma_n\}$ in the data source taxonomies, through the means of user-defined mappings Ψ. The abstract instance space defined by Γ^U is denoted by I_{Γ^U} and is given by $I_{\Gamma_U} = \varphi_1(I_{\Gamma_1}) \cup \varphi_2(I_{\Gamma_2}) ... \cup \varphi_p(I_{\Gamma_n})$. The set of user AVT $O_U = T_1^U, ..., T_n^U$ induces an instance space $I_{O_u} = \cup_{\Gamma^U} I_{\Gamma^U}$. We say that an instance $x \in I_{O_U}$ is *partially specified* if one of more of its attribute values are not primitive. A *partially specified data set* D_{O_U} (relative to a set O_U of user AVT) is a collection of instances drawn from I_{O_U} where each instance is labeled with the appropriate class label from $C = \{c_1, c_2, \cdots, c_k\}$, a finite set of mutually disjoint classes. Thus, $D_{O_U} \subset I_{O_U} \times C$.

The problem of learning classifiers from distributed, semantically heterogeneous data sources can be formulated as follows: Given a collection of (possibly) partially specified data sources $D_1, \cdots, D_p$ and their associated collections of taxonomies $\{O_1, ..., O_p\}$, a user collection of taxonomies O_U and a set of mappings Ψ from data source taxonomies O_i to user taxonomies O_U, construct a classifier $h_{O_U} : I_{O_U} \rightarrow C$ for assigning appropriate class labels to each instance in the instance space I_{O_U}.

4.4 AVT-Guided Learning Algorithms

AVT-guided learning algorithms extend standard learning algorithms in principled ways so as to exploit the information provided by AVT. We have designed and implemented AVT-NBL [74] and AVT-DTL [73] for learning AVT-guided Naive Bayes and Decision Tree classifiers, respectively. The standard Decision Trees or Naive Bayes learning algorithms can be viewed as special cases of AVT-DTL or AVT-NBL, where the AVT associated with each attribute has only one level. The root of such an AVT corresponds to the value of the attribute being unknown and the leaves correspond to the primitive values of the attribute. We will use Naive Bayes Learner (NBL) as an example to illustrate our approach to AVT-guided learning algorithms. Naive Bayes classifier operates under the assumption that each attribute is independent of the others given the class. Thus, the joint class conditional probability of an instance can be written as the product of individual class conditional probabilities corresponding to each attribute defining the instance. The Bayesian approach to classifying an instance $x = \{v_1, \cdots, v_n\}$ is to assign it to the most probable class $c_{MAP}(x)$. We have: $c_{MAP}(x) = \underset{c_j \in C}{\operatorname{argmax}}\, p(v_1, \cdots, v_n | c_j) p(c_j) = \underset{c_j \in C}{\operatorname{argmax}}\, p(c_j) \prod_i p(v_i | c_j)$.

Therefore, the task of the Naive Bayes Learner (NBL) is to estimate the class probabilities $p(c_j)$ and the class conditional probabilities $p(v_i|c_j)$, for all classes $c_j \in \mathbf{C}$ and for all attribute values $v_i \in dom(A_i)$. These probabilities can be estimated from a training set D using standard probability estimation methods [1] based on relative frequency counts. We denote by $\sigma(v_i|c_j)$ the frequency count of a value v_i of the attribute A_i given the class label c_j, and by $\sigma(c_j)$ the frequency count of the class label c_j in a training set D.

AVT-guided NBL, called AVT-NBL [74] efficiently exploits taxonomies defined over values of each attribute in a data set to find a Naive Bayes classifier that optimizes the Conditional Minimum Description Length (CMDL) score [Friedman et al., 1997]. More precisely, the task of AVT-NBL is to construct a Naive Bayes classifier for assigning an unlabeled instance $x \in I_O$ to its most probable class $c_{MAP}(x)$. As in the case of NBL, we assume that each attribute is independent of the other attributes given the class. A Naive Bayes classifier defined on the instance space I_O is completely specified by a set of class conditional probabilities for each value of each attribute. Suppose we denote the table of class conditional probabilities associated with values in γ_i by $CPT(\gamma_i)$. Then the Naive Bayes classifier defined over the instance space I_O is specified by $h(\Gamma) = \{CPT(\gamma_1), \cdots, CPT(\gamma_n)\}$.

AVT-NBL starts with the Naive Bayes Classifier that is based on the most abstract value of each attribute (the most general hypothesis) and successively refines the classifier (hypothesis) using a criterion that is designed to tradeoff between the accuracy of classification and the complexity of the resulting Naive Bayes classifier. Successive refinements of Γ correspond to an ordering of Naive Bayes classifiers based on the structure of the AVTs in O. For example, in Figure 8, Γ' is a refinement of Γ, and hence the corresponding hypothesis $h(\Gamma')$ is a refinement of $h(\Gamma)$ [74].

The scoring function that we use to evaluate a candidate AVT-guided refinement of a Naive Bayes Classifier is an adaptation (for the case of classifiers constructed from partially specified data) of the Conditional Minimum Description Length (CMDL) criterion [76] and captures the tradeoff between the accuracy and the complexity of the resulting Naive Bayes classifier [74].

The parameters that define the classifier can be estimated from the observed class distribution in the data D based on frequency counts $\sigma_i(c_j))$ and $p_i(v|c_j)$ is the class conditional probability of value v of attribute A_i given the class label c_j.

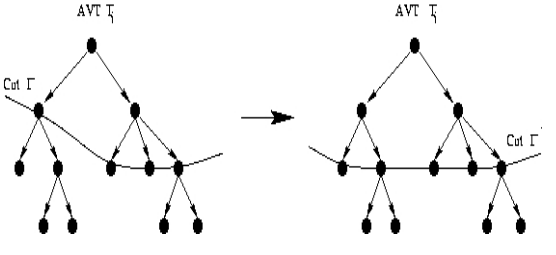


Fig. 8. Global cut through a set of attribute value taxonomies

The value of $p_i(v|c_j)$ can similarly estimated from the frequency counts $\sigma_i(v|c_j))$ obtained from the data set D. When some of the data are partially specified, we can use a 2-step process for computing $\sigma_i(v|c_j))$. First we make an upward pass through the AVT for each attribute aggregating the class conditional frequency counts based on the specified attribute values in the data set; Then we propagate the counts associated with partially specified attribute values down through the tree, augmenting the counts at lower levels according to the distribution of values along the branches based on the subset of the data for which the corresponding values are fully specified [73, 74]. This procedure can be seen as a special case of EM (Expectation Maximization) algorithm for estimation of $\sigma_i(v|c_j)$ under the assumption that the attributes are independent given the class [74].

Thus, AVT-NBL produces a hypothesis h that intuitively trades off the complexity of Naive Bayes classifier (in terms of the number of parameters used to describe the relevant class conditional probabilities) against accuracy of classification. The algorithm terminates when none of the candidate refinements of the classifier yield statistically significant improvement in the $CMDL$ score [74]. Our experiments with several synthetic as well as real-world data sets have demonstrated the efficacy of AVT-NBL [74] and AVT-DTL [73].

4.5 Learning Classifiers from Partially Specified Semantically Heterogeneous Data

Our approach to AVT-guided learning from partially specified semantically heterogeneous data [77] relies on our general strategy for transforming algorithms for learning from data into algorithms for learning from distributed, semantically heterogeneous data [11, 20]. As mentioned before, this strategy is based on the decomposition of the learning task into an information extraction component (when *sufficient statistics* needed for learning are gathered) and a hypothesis generation component (that uses the *sufficient statistics* to generate or refine a current hypothesis).

Recall that a statistic $s_L(D)$ is a *sufficient statistic for learning* a hypothesis h using a learning algorithm L applied to a data set D if there exists a procedure that takes $s_L(D)$ as input and outputs h [20]. For example, in the case of NBL, the frequency counts $\sigma(v_i|c_j)$ of the value v_i of the attribute A_i given the class label c_j in a training set D, and the frequency count $\sigma(c_j)$ of the class label c_j in a training set D completely summarize the information needed for constructing a Naive Bayes classifier from D, and thus, they constitute *sufficient statistics* for NBL. As noted in Section 2, some simple learning algorithms such as NBL $s_L(D)$, the sufficient statistics required for constructing the classifier can be computed in one step, in general, a learning algorithm may require assembly of $s_L(D)$ through an interleaved execution of the information extraction and hypothesis generation components [20].

We illustrate our approach to using this strategy to design AVT-guided algorithms for learning classifiers from semantically heterogeneous data using the Naive Bayes classifier as an example. However, the proposed approach can be extended to a broad range of machine learning algorithms including variants

of Decision Tree, Bayesian networks (Naive Bayes and Tree-Augmented Naive Bayes classifiers), generalized linear models, support vector machines.

Sufficient Statistics for AVT-NBL. As we have shown, AVT-NBL starts with a Naive Bayes classifier $h_0 = h(\Gamma_0)$ corresponding to the most abstract cut Γ_0 in the attribute value taxonomy associated with the data (i.e., the most general classifier that simply assigns each instance to the class that is apriori most probable) and it iteratively refines the classifier by refining the corresponding cut until a best cut, according to the performance criterion, is found. More precisely, let h_i be the current hypothesis corresponding to the current cut Γ (i.e., $h_i = h(\Gamma)$) and Γ' a (one-step) refinement of Γ (see Figure 8).

Let $h(\Gamma')$ be the Naive Bayes classifier corresponding to the cut Γ' and let $CMDL(\Gamma|D)$ and $CMDL(\Gamma'|D)$ be the CMDL scores corresponding to the hypotheses $h(\Gamma)$ and $h(\Gamma')$, respectively. If $CMDL(\Gamma) > CMDL(\Gamma')$ then $h_{i+1} = h(\Gamma')$, otherwise $h_{i+1} = h(\Gamma)$. This procedure is repeated until no (one-step) refinement Γ' of the cut Γ results in a significant improvement of the CMDL score, and the algorithm ends by outputing the classifier $h(\Gamma)$.

Thus, the classifier that the AVT-NBL finds is obtained from $h_0 = h(\Gamma_0)$ through a sequence of refinement operations. The refinement sufficient statistics $s_L(D, h_i \rightarrow h_{i+1})$ are identified below.

Let h_i be the current hypothesis corresponding to a cut Γ and $CMDL(\Gamma|D)$ its score. If Γ' is a refinement of the cut Γ, then the refinement sufficient statistics needed to construct h_{i+1} are given by the frequency counts needed to construct $h(\Gamma')$ together with the probabilities needed to compute $CLL(h(\Gamma')|D)$ (calculated once we know $h(\Gamma')$). If we denote by $dom_{\Gamma'}(A_i)$ the domain of the attribute A_i when the cut Γ' is considered, then the frequency counts needed to construct $h(\Gamma')$ are $\sigma(v_i|c_j)$ for all values $v_i \in dom_{\Gamma'}(A_i)$ of all attributes A_i and for all class values $c_j \in dom_{\Gamma'}(C)$, and $\sigma(c_j)$ for all class values $c_j \in dom_{\Gamma'}(C)$. To compute $CLL(h(\Gamma')|D)$ the products $\prod_j p_{h(\Gamma')}(v_{ij}|c_k)$ for all examples $x_i = (v_{i1}, \cdots, v_{in})$ and for all classes $c_k \in C$ are needed.

The step $i+1$ of the algorithm corresponding to the cut Γ' can be briefly described in terms of information gathering and hypothesis generation components as follows:

(1) Compute $\sigma(v_i|c_j)$ and $\sigma(c_j)$ corresponding to the cut Γ' from the training data D
(2) Generate the NB classifier $h(\Gamma')$
(3) Compute $\prod_j p_{h(\Gamma')}(v_{ij}|c_k)$ from D
(4) Generate the hypothesis h_{i+1}

Learning Naive Bayes Classifiers from Semantically Heterogeneous Data. Let $\{D_1, \cdots, D_p\}$ be a set of semantically heterogeneous data sources with associated ontologies $\{O_1, \cdots O_p\}$. Let O_U be a user collection of AVT and Γ a cut through the user AVT.

The step $i + 1$ (corresponding to the cut Γ' in the user ontology) of the algorithm for learning Naive Bayes classifiers from distributed, semantically heterogeneous data sources $D_1, \cdots, D_p$ can be described in terms of information gathering and hypothesis generation components as follows:

(1) Compute $\sigma(v_i|c_j)$ and $\sigma(c_j)$ corresponding to the cut Γ' from the distributed data sources $D_1, \cdots, D_p$
(2) Generate the NB classifier $h(\Gamma')$ at the user location and send it to the data sources $D_1, \cdots, D_p$
(3) Compute $\prod_j p_{h(\Gamma')}(v_{ij}|c_k)$ from $D_1, \cdots, D_p$
(4) Generate the hypothesis h_{i+1} at the user location

Thus, using the decomposition of an AVT-guided algorithm for learning classifier from partially specified data into information extraction and hypothesis generation components, we reduce the problem of learning classifiers from distributed, semantically heterogeneous data sources to the problem of querying for sufficient statistics from such data sources (e.g., frequency counts $\sigma(v_i|c_j)$ and $\sigma(c_j)$ corresponding to a cut). This involves design of procedures for decomposing statistical queries into sub-queries corresponding to the distributed data sources and procedures for combining the partial answers into a final answer to the initial queries (e.g., adding up counts) [77].

4.6 Related Work on Learning Classifiers from Partially Specified Data

Walker [78] first used attribute value taxonomies in information retrieval from large databases. DeMichiel [79], and Chen and Tseng [80] proposed database models to handle imprecision using partial values and associated probabilities where a partial value refers to a set of possible values for an attribute. McClean et al. [81] proposed aggregation operators defined over partial values. While this work suggests ways to aggregate statistics so as to minimize information loss, it does not address the problem of learning from AVT and partially specified data. The problem of learning classifiers from AVT and partially specified data was formulated and solved in the case of decision tree classifiers by Zhang and Honavar [73] and in the case of Naive Bayes classifiers by Zhang and Honavar [74]. Development of approaches to exploit abstractions over attribute values and class labels to optimally exploit partially specified data. The use of prior knowledge or domain theories specified typically in first order logic or propositional logic to guide learning from data has been explored in ML-SMART [82], FOCL [83] and KBANN [84] systems as well as in the work of Aronis et al. [85] and Aronis and Provost [86]. However, the work on exploiting domain theories in learning has not focused on the effective use of AVT to learn classifiers from partially specified data. Approaches to exploiting abstractions in learning from fully specified data have been studied by several authors [87, 88, 89, 90, 91, 92, 93, 94, 95]. We have developed simple algorithms for learning decision tree [73] and Naive Bayes [74] classifiers from partially specified data. These methods assume independence of attributes in estimating answers to statistical queries from partially specified data based on the distribution of observed values. in fully specified instances. It is also of interest to investigate methods based on multiple imputation [96, 97, 98] which has been used with success in a number of applications such as studies of air quality [99], employment [100], and health care [101] to cope with missing observations. Multiple imputation aims to: (a) use available information to make

good predictions of the missing values and (b) reflect uncertainty due to the fact that some of the data were in fact not observed. Some causes of missing data such as when an individual does not answer a particular question, and when an individual refuses to answer any questions, but some demographic information such as the identity of the data source that the person is associated with is available, have been considered in the statistical literature [102, 103, 104].

5 Summary and Discussion

Biological, environmental, ecological, engineering, social, and biomedical sciences are in the midst of being transformed from data poor sciences into data rich sciences, in large part, due to rapid advances in experimental and data acquisition methods. Recent advances in computer science, statistical methods, and information theory provide powerful conceptual tools for extracting knowledge from data and for developing algorithmic models of causal interactions within and across multiple levels of organization in complex systems. Advances in computing, storage, communication, and software technologies (e.g., web services that can be invoked and on the Internet and executed on remote computers or data repositories) provide unprecedented opportunities for exploiting disparate data and knowledge to address fundamental scientific questions. Because data sources that are created for use by one scientific community (e.g., structural biologists) find use in other contexts (e.g. exploration of macromolecular function), given the prevalence of discipline-specific terminologies (ontologies), semantic differences among autonomous data repositories are simply unavoidable. Effective use of multiple sources of data in a given context requires reconciliation of such semantic differences from the user's point of view. This is especially true in emerging areas of scientific inquiry at the boundaries of established disciplines (e.g., computational biology) that draw on multiple areas of inquiry (e.g., molecular biology, biophysics, structural biology). Furthermore, because many of the data sources of interest are autonomous and geographically distributed, it is neither desirable nor feasible to gather all of the data in a centralized location for analysis. Hence, there is an urgent need for algorithms and software for collaborative discovery from autonomous, semantically heterogeneous, distributed information sources. Against this background, the research summarized in this paper has led to:

(a) The development of a general theoretical framework for learning predictive models (e.g., classifiers) from large, physically distributed data sources where it is neither desirable nor feasible to gather all of the data in a centralized location for analysis [20]. This framework offers a general recipe for the design of algorithms for learning from distributed data that are provably exact with respect to their centralized counterparts (in the sense that the model constructed from a collection of physically distributed data sets is provably identical to that obtained in the setting where the learning algorithm has access to the entire data set). A key feature of this framework is the clear separation of concerns between hypothesis construction and extraction and

refinement of sufficient statistics needed by the learning algorithm from data which reduces the problem of learning from data to a problem of decomposing a query for sufficient statistics across multiple data sources and combining the answers returned by the data sources to obtain the answer for the original query. This work has resulted in the identification of sufficient statistics for a large family of learning algorithms including in particular, algorithms for learning decision trees [20], neural networks, support vector machines [23] and Bayesian networks, and consequently, provably exact algorithms for learning the corresponding classifiers from distributed data.

(b) The development of theoretically sound yet practical variants of a large class of algorithms [20, 23] for learning predictive models (classifiers) from distributed data sources under a variety of assumptions (motivated by practical applications) concerning the nature of data fragmentation, and the query capabilities and operations permitted by the data sources (e.g., execution of user supplied procedures), and precise characterization of the complexity (computation, memory, and communication requirements) of the resulting algorithms relative to their centralized counterparts.

(c) The development of a theoretically sound approach to formulation and execution of statistical queries across semantically heterogeneous data sources [11]. This work has demonstrated how to use semantic correspondences and mappings specified by users from a set of terms and relationships among terms (user ontology) to terms and relations in data source specific ontologies to construct a sound procedure for answering queries for sufficient statistics needed for learning classifiers from semantically heterogeneous data. An important component of this work has to do with the development of statistically sound approaches to learning classifiers from *partially specified data* resulting from data described at different levels of abstraction across different data sources [73, 74].

(d) The design and implementation of INDUS, a modular, extensible, open-source software toolkit (`http://www.cild.iastate.edu/software/indus.html`) for data-driven knowledge acquisition from large, distributed, autonomous, semantically heterogeneous data sources [44, 11].

(e) Applications of the resulting approaches to data-driven knowledge acquisition tasks that arise in bioinformatics [30, 44, 105, 106].

Work in progress is aimed at:

(a) Extending the INDUS query answering engine to flexibly interact with different data sources that might support different functionalities or impose different constraints on users (For example, some data sources might answer only restricted classes of statistical queries. Others might allow retrieval of raw data. Still others might allow execution of user-supplied procedures at the data source, there by allowing the users to effectively extend the query capabilities of the data source);

(b) Investigation of resource-bounded approximations of answers to statistical queries generated by the learner; develop approximation criteria for evaluation of the quality of classifiers obtained in the distributed setting under

a given set of resource constraints and query capabilities relative to that obtained in the centralized setting with or without such constraints. This is especially important in application scenarios in which it is not feasible to obtain exact answers to statistical queries posed under the access and resource constraints imposed by the distributed setting;

(c) Development of tools to support modular development of ontologies, interactive specification of mappings between ontologies including automated generation of candidate mappings for consideration by users, and reasoning algorithms for ensuring semantic integrity of user-specified mappings between ontologies;

(d) Development of sophisticated approaches to estimation from partially specified data, of the statistics needed by learning algorithms; and

(e) Application of the resulting algorithms and software to collaborative discovery problems that arise in areas such as computational biology e.g., discovery of relationships between macromolecular sequence, structure, expression, interaction, function, and evolution; discovery of genetic regulatory networks from multiple sources of data (e.g., gene expression, protein localization, protein-protein interaction).

Acknowledgements

This research was supported in part by grants from the National Science Foundation (NSF IIS 0219699) and the National Institutes of Health (GM 066387) to Vasant Honavar. This work has benefited from discussions with Adrian Silvescu, Jaime Reinoso-Castillo, and Drena Dobbs.

References

[1] Mitchell, T.: Machine Learning. McGraw Hill (1997)
[2] Duda, R., Hart, E., Stork, D.: Pattern Recognition. Wiley (2000)
[3] Thrun, S., Faloutsos, C., Mitchell, M., Wasserman, L.: Automated learning and discovery: State-of-the-art and research topics in a rapidly growing field. AI Magazine (1999)
[4] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning : Data Mining, Inference, and Prediction. Springer-Verlag (2001)
[5] Bishop, C.M.: Neural Networks for Pattern Recognition. New York: Oxford University Press (1995)
[6] Baldi, P., Frasconi, P., Smyth, P.: Modeling the Internet and the Web - Probabilistic Methods and Algorithms. New York: Wiley (2003)
[7] Baldi, P., Brunak, S.: Bioinformatics - A Machine Learning Approach. Cambridge, MA: MIT Press (2003)
[8] Sowa, J.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. New York: PWS Publishing Co. (1999)
[9] Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., Harris, M., Hill, D., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J., Richardson, J., Ringwald, M., Rubin, G., Sherlock, G.: Gene ontology: tool for unification of biology. Nature Genetics **25** (2000) 25–29

[10] Reinoso-Castillo, J., Silvescu, A., Caragea, D., Pathak, J., Honavar, V.: Information extraction and integration from heterogeneous, distributed, autonomous information sources: a federated, query-centric approach. In: IEEE International Conference on Information Integration and Reuse, Las Vegas, Nevada (2003)
[11] Caragea, D., Pathak, J., Honavar, V.: Learning classifiers from semantically heterogeneous data. In: Proceedings of the International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems. (2004)
[12] Dzeroski, S., Lavrac, N., eds.: Relational Data Mining. Springer-Verlag (2001)
[13] Getoor, L., Friedman, N., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In Dzeroski, S., N. Lavrac, E., eds.: Relational Data Mining. Springer-Verlag (2001)
[14] Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, Orlando, FL, Morgan Kaufmann Publishers Inc. (1999) 1300–1309
[15] Atramentov, A., Leiva, H., Honavar, V.: Learning decision trees from multi-relational data. In Horvth, T., Yamamoto, A., eds.: Proceedings of the 13th International Conference on Inductive Logic Programming. Volume 2835 of Lecture Notes in Artificial Intelligence., Springer-Verlag (2003) 38–56
[16] Neville, J., Jensen, D., Gallagher, B.: Simple estimators for relational bayesian classifiers. In: ICDM 2003. (2003)
[17] Casella, G., Berger, R.: Statistical Inference. Duxbury Press, Belmont, CA (2001)
[18] Davidson, A.: Statistical Models. London: Cambridge University Press (2003)
[19] Kearns, M.: Efficient noise-tolerant learning from statistical queries. Journal of the ACM **45** (1998) 983–1006
[20] Caragea, D., Silvescu, A., Honavar, V.: A framework for learning from distributed data using sufficient statistics and its application to learning decision trees. International Journal of Hybrid Intelligent Systems **1** (2004)
[21] Caragea, D., Silvescu, A., Honavar, V.: Decision tree induction from distributed heterogeneous autonomous data sources. In: Proceedings of the International Conference on Intelligent Systems Design and Applications, Tulsa, Oklahoma (2003)
[22] Caragea, D., Silvescu, A., Honavar, V.: Agents that learn from distributed dynamic data sources. In: Proceedings of the Workshop on Learning Agents, Agents 2000/ECML 2000, Barcelona, Spain (2000) 53–61
[23] Caragea, C., Caragea, D., , Honavar, V.: Learning support vector machine classifiers from distributed data. extended abstract. In: Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI 2005). (2005)
[24] Caragea, D.: Learning classifiers from Distributed, Semantically Heterogeneous, Autonomous Data Sources. Ph.d. thesis, Department of Computer Science. Iowa State University, Ames, Iowa, USA (2004)
[25] Quinlan, R.: Induction of decision trees. Machine Learning **1** (1986) 81–106
[26] Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and regression trees. Wadsworth, Monterey, CA (1984)
[27] Graefe, G., Fayyad, U., Chaudhuri, S.: On the efficient gathering of sufficient statistics for classification from large sql databases. In: Proceedings of the Fourth International Conference on KDD, Menlo Park, CA, AAAI Press (1998) 204–208
[28] Moore, A.W., Lee, M.S.: Cached sufficient statistics for efficient machine learning with large datasets. Journal of Artificial Intelligence Research **8** (1998) 67–91

[29] Wang, X., Schroeder, D., Dobbs, D., , Honavar, V.: Data-driven discovery of rules for protein function classification based on sequence motifs: Rules discovered for peptidase families based on meme motifs outperform those based on prosite patterns and profiles. In: Proceedings of the Conference on Computational Biology and Genome Informatics. (2002)
[30] Andorf, C., Silvescu, A., Dobbs, D., Honavar, V.: Learning classifiers for assigning protein sequences to gene ontology functional families. In: Fifth International Conference on Knowledge Based Computer Systems (KBCS 2004), India (2004)
[31] Cortes, C., Vapnik, V.: Support vector networks. Machine Learning **20** (1995) 273–297
[32] Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press (2000)
[33] Bradley, P.S., Mangasarian, O.L.: Massive data discrimination via linear support vector machines. Optimization Methods and Software **13(1)** (2000) 1–10
[34] Srivastava, A., Han, E., Kumar, V., Singh, V.: Parallel formulations of decision-tree classification algorithms. Data Mining and Knowledge Discovery **3** (1999) 237–261
[35] Grossman, L., Gou, Y.: Parallel methods for scaling data mining algorithms to large data sets. In Zytkow, J., ed.: Handbook on Data Mining and Knowledge Discovery. Oxford University Press (2001)
[36] Provost, F.J., Kolluri, V.: A survey of methods for scaling up inductive algorithms. Data Mining and Knowledge Discovery **3** (1999) 131–169
[37] Park, B., Kargupta, H.: Constructing simpler decision trees from ensemble models using Fourier analysis. In: Proceedings of the 7th Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'2002), Madison, WI, ACM SIGMOD (2002) 18–23
[38] Domingos, P.: Knowledge acquisition from examples via multiple models. In: Proceedings of the Fourteenth International Conference on Machine Learning, Nashville, TN, Morgan Kaufmann (1997) 98–106
[39] Prodromidis, A., Chan, P., Stolfo, S.: Meta-learning in distributed data mining systems: issues and approaches. In Kargupta, H., Chan, P., eds.: Advances of Distributed Data Mining. AAAI Press (2000)
[40] Bhatnagar, R., Srinivasan, S.: Pattern discovery in distributed databases. In: Proceedings of the Fourteenth AAAI Conference, Providence, RI, AAAI Press/The MIT Press (1997) 503–508
[41] Kargupta, H., Park, B., Hershberger, D., Johnson, E.: Collective data mining: A new perspective toward distributed data mining. In Kargupta, H., Chan, P., eds.: Advances in Distributed and Parallel Knowledge Discovery. MIT Press (1999)
[42] Mansour, J.: Learning boolean functions via the fourier transform. In: Theoretical Advances in Neural Computation and Learning. Kluwer (1994)
[43] Levy, A.: Logic-based techniques in data integration. In: Logic-based artificial intelligence. Kluwer Academic Publishers (2000) 575–595
[44] Caragea, D., Silvescu, A., Pathak, J., Bao, J., Andorf, C., Dobbs, D., Honavar, V.: Information integration and knowledge acquisition from semantically heterogeneous biological data sources. In: Proceedings of the Second International Workshop on Data Integration in Life Sciences, (DILS 2005), San Diego, CA, Berlin: Springer-Verlag. Lecture Notes in Computer Science (2005)
[45] Bonatti, P., Deng, Y., Subrahmanian, V.: An ontology-extended relational algebra. In: Proceedings of the IEEE Conference on Information Integration and Reuse, IEEE Press (2003) 192–199

[46] Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: C-owl: Contextualizing ontologies. In: Proceedings of the Second International Semantic Web Conference, Springer Verlag, LNCS 2870 (2003)
[47] Bao, J., Honavar, V.: Collaborative ontology building with wiki@nt - a multi-agent based ontology building environment. In: Proceedings of the Third International Workshop on Evaluation of Ontology based Tools, at the Third International Semantic Web Conference ISWC, Hiroshima, Japan (2004)
[48] Bao, J., Honavar, V.: An efficient algorithm for reasoning about subsumption and equivalence relationships to support collaborative editing of ontologies and inter-ontology mappings. under review. (2005)
[49] Hull, R.: Managing semantic heterogeneity in databases: A theoretical perspective. In: PODS, Tucson, Arizona (1997) 51–61
[50] Davidson, S., Crabtree, J., Brunk, B., Schug, J., Tannen, V., Overton, G., Stoeckert, C.: K2/Kleisli and GUS: experiments in integrated access to genomic data sources. IBM Journal **40** (2001)
[51] Eckman, B.: A practitioner's guide to data management and data integration in bioinformatics. Bioinformatics (2003) 3–74
[52] Sheth, A., Larson, J.: Federated databases: architectures and issues. ACM Computing Surveys **22** (1990) 183–236
[53] Barsalou, T., Gangopadhyay, D.: M(dm): An open framework for interoperation of multimodel multidatabase systems. IEEE Data Engineering (1992)
[54] Bright, M., Hurson, A., Pakzad, S.: A taxonomy and current issues in multibatabase systems. Computer Journal **25** (1992) 5–60
[55] Wiederhold, G., Genesereth, M.: The conceptual basis for mediation services. IEEE Expert **12** (1997) 38–47
[56] Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J., Vassalos, V., Widom, J.: The TSIMMIS approach to mediation: data models and languages. Journal of Intelligent Information Systems, Special Issue on Next Generation Information Technologies and Systems **8** (1997)
[57] Chang, C.K., Garcia-Molina, H.: Mind your vocabulary: query mapping across heterogeneous information sources. In: ACM SIGMOD International Conference On Management of Data, Philadelphia, PA (1999)
[58] Arens, Y., Chin, C., Hsu, C., Knoblock, C.: Retrieving and integrating data from multiple information sources. International Journal on Intelligent and Cooperative Information Systems **2** (1993) 127–158
[59] Knoblock, C., Minton, S., Ambite, J., Ashish, N., Muslea, I., Philpot, A., Tejada, S.: The ariadne approach to Web-based information integration. International Journal of Cooperative Information Systems **10** (2001) 145–169
[60] Lu, J., Moerkotte, G., Schue, J., Subrahmanian, V.: Efficient maintenance of materialized mediated views. In: Proceedings of 1995 ACM SIGMOD Conference on Management of Data, San Jose, CA (1995)
[61] Levy, A.: The information manifold approach to data integration. IEEE Intelligent Systems **13** (1998)
[62] Draper, D., Halevy, A.Y., Weld, D.S.: The nimble XML data integration system. In: ICDE. (2001) 155–160
[63] Etzold, T., Harris, H., Beulah, S.: SRS: An integration platform for databanks and analysis tools in bioinformatics. Bioinformatics Managing Scientific Data (2003) 35–74
[64] Haas, L., Schwarz, P., Kodali, P., Kotlar, E., Rice, J., Swope, W.: DiscoveryLink: a system for integrated access to life sciences data sources. IBM System Journal **40** (2001)

[65] Stevens, R., Goble, C., Paton, N., Becchofer, S., Ng, G., Baker, P., Bass, A.: Complex query formulation over diverse sources in tambis. Bioinformatics (2003) 189–220
[66] Chen, J., Chung, S., Wong, L.: The Kleisli query system as a backbone for bioinformatics data integration and analisis. Bioinformatics (2003) 147–188
[67] Tannen, V., Davidson, S., Harker, S.: The information integration in K2. Bioinformatics (2003) 225–248
[68] Tomasic, A., Rashid, L., Valduriez, P.: Scaling heterogeneous databases and design of DISCO. IEEE Transactions on Knowledge and Data Engineering **10** (1998) 808–823
[69] Haas, L., Kossmann, D., Wimmers, E., Yan, J.: Optimizing queries across diverse sources. In: Proceedings of the 23rd VLDB Conference, Athens, Greece (1997) 267–285
[70] Rodriguez-Martinez, M., Roussopoulos, R.: MOCHA: a self-extensible database middleware system for distributed data sources. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX (2000) 213–224
[71] Lambrecht, E., Kambhampati, S., Gnanaprakasam, S.: Optimizing recursive information-gathering plans. In: Proceedings of the International Joint Conference on Artificial Intelligence, AAAI Press (1999) 1204–1211
[72] Maluf, D., Wiederhold, G.: Abstraction of representation in interoperation. Lecture Notes in AI **1315** (1997)
[73] Zhang, J., Honavar, V.: Learning decision tree classifiers from attribute-value taxonomies and partially specified data. In Fawcett, T., Mishra, N., eds.: Proceedings of the International Conference on Machine Learning, Washington, DC (2003) 880–887
[74] Zhang, J., Honavar, V.: Learning concise and accurate naive bayes classifiers from attribute value taxonomies and data. In: Proceedings of the Fourth ICMD. (2004)
[75] Haussler, D.: Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. Artificial Intelligence **36** (1988) 177–221
[76] Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning **29** (1997)
[77] Caragea, D., Zhang, J., Pathak, J., Honavar, V.: Learning classifiers from distributed, ontology-extended data sources. under review. (2005)
[78] Walker, A.: On retrieval from a small version of a large database. In: VLDB Conference, 1989. (1989)
[79] DeMichiel, L.: Resolving database incompatibility: An approach to performing relational operations over mismatched domains. IEEE Trans. Knowl. Data Eng. **1** (1989)
[80] Chen, A., Tseng, F.: Evaluating aggregate operations over imprecise data. IEEE Trans. On Knowledge and Data Engineering **8** (1996)
[81] McClean, S., Scotney, B., Shapcott, M.: Aggregation of imprecise and uncertain information in databases. IEEE Transactions on Knowledge and Data Engineering **6** (2001)
[82] Bergadano, F., Giordana, A.: Guiding induction with domain theories. In: Machine Learning An Artificial Intelligence Approach. Volume 3. Palo Alto, CA: Morgan Kaufmann (1990)
[83] Pazzani, M., Kibler, D.: The role of prior knowledge in inductive learning. Machine Learning **9** (1992)

[84] Towell, G., Shavlik, J.: Knowledge-based artificial neural networks. Artificial Intelligence **70** (1994)
[85] Aronis, J., Kolluri, V., Provost, F., Buchanan, B.: The WoRLD: knowledge discovery from multiple distributed databases. Technical Report ISL-96-6, Intelligent Systems Laboratory, Department of Computer Science, University of Pittsburgh, Pittsburgh, PA (1996)
[86] Aronis, J., Provost, F.: Increasing the efficiency of inductive learning with breadth-first marker propagation. In: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining. (1997)
[87] Nunez, M.: The use of background knowledge in decision tree induction. Machine Learning **6** (1991)
[88] H., A., Akiba, Y., Kaneda, S.: On handling tree-structured attributes. In: Proceedings of the Twelfth International Conference on Machine Learning. (1995)
[89] Dhar, V., Tuzhilin, A.: Abstract-driven pattern discovery in databases. IEEE Transactions on Knowledge and Data Engineering **5** (1993)
[90] Han, J., Fu, Y.: Exploration of the power of attribute-oriented induction in data mining, u.m. fayyad el al. (eds.). In: Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press (1996)
[91] Hendler, J., Stoffel, K., , Taylor, M.: Advances in high performance knowledge representation (1996)
[92] Taylor, M., Stoffel, K., Hendler, J.: Ontology-based induction of high level classification rules. In: SIGMOD Data Mining and Knowledge Discovery workshop proceedings, Tuscon, Arizona (1997)
[93] Pazzani, M., Mani, S., Shankle, W.: Beyond concise and colorful: Learning intelligible rules. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA (1997)
[94] Pazzani, M., Mani, M., Shankle, W.: Comprehensible knowledge discovery in databases. In: Proceedings of the the Cognitive Science Conference. (1997)
[95] desJardins, M., Getoor, L., Koller, D.: Using feature hierarchies in bayesian network learning. In: Proceedings of the Symposium on Abstraction, Reformulation, Approximation. Lecture Notes in Artificial Intelligence 1864: 260-270, Springer-Verlag (2000)
[96] Rubin, D.: Multiple imputations in sample surveys: A phenomenological bayesian approach to nonresponse (c/r: p29-34). In: Proceedings of the American Statistical Association, Section on Survey Research Methods. ((1978))
[97] Rubin, D.: Multiple imputation for nonresponse in surveys. John Wiley and Sons (New York; Chichester) (1987)
[98] Rubin, D.: Multiple imputation after 18+ years. Journal of the American Statistical Association **91** (1996)
[99] Junninen, H., Niska, H., Tuppurainen, K., Ruuskanen, J., , Kolehmainen, M.: Methods for imputation of missing values in air quality data sets. Atmospheric Environment **38** (2004)
[100] Longford, N.: Missing data and small area estimation in the uk labour force survey. Journal of the Royal Statistical Society Series A-Statistics in Society **167** (2004)
[101] Raghunathan, T.: What do we do with missing data? some options for analysis of incomplete data. Annual Review of Public Health **25** (2004)
[102] Little, R., Rubin, D.: Statistical analysis with missing data. John Wiley and Sons (New York; Chichester), 2nd edition (2002)
[103] Madow, W., Olkin, I., , Rubin, D.B., e.: Incomplete data in sample surveys (Vol. 2): Theory and bibliographies. Academic Press (New York; London) (1983)

[104] Madow, W., Nisselson, J., Olkin, I.e.: Incomplete data in sample surveys (Vol. 1): Report and case studies. Academic Press (New York; London) (1983)
[105] Yan, C., Dobbs, D., Honavar, V.: A two-stage classifier for identification of protein-protein interface residues. Bioinformatics **20** (2004) i371–378
[106] Yan, C., Honavar, V., Dobbs, D.: Identifying protein-protein interaction sites from surface residues - a support vector machine approach. Neural Computing Applications **13** (2004) 123–129