Learning Classifiers from Distributed, Ontology-Extended Data Sources

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

AI Research Lab, Department of Computer Science, Iowa State University
226 Atanasoff Hall, Ames, IA 50011
{dcaragea, zhang, jpathak, honavar}@cs.iastate.edu

Abstract. There is an urgent need for sound approaches to integrative and collaborative analysis of large, autonomous (and hence, inevitably semantically heterogeneous) data sources in several increasingly data-rich application domains. In this paper, we precisely formulate and solve the problem of learning classifiers from such data sources, in a setting where each data source has a hierarchical ontology associated with it and semantic correspondences between data source ontologies and a user ontology are supplied. The proposed approach yields algorithms for learning a broad class of classifiers (including Bayesian networks, decision trees, etc.) from semantically heterogeneous distributed data with strong performance guarantees relative to their centralized counterparts. We illustrate the application of the proposed approach in the case of learning Naive Bayes classifiers from distributed, ontology-extended data sources.

1 Introduction

The availability of large amounts of data in many application domains has resulted in unprecedented opportunities for data driven knowledge discovery. However, the massive size, the distributed nature of the data sources and the inevitability of semantic differences between independently managed data repositories present significant hurdles in our ability to fully exploit such data sources in knowledge discovery. The Semantic Web enterprise [1] is aimed at supporting seamless and flexible access and use of semantically heterogeneous data sources by associating meta-data (e.g., ontologies) with data available in many application domains. The work described in this paper is aimed at the development of algorithms for learning concise and accurate classifiers from semantically heterogeneous, distributed data sets for applications in which integration of data from multiple sources into a centralized repository is not feasible (e.g., because of the enormous size of the data sources).

The problem that we seek to address is best illustrated by an example: Consider two academic departments that independently collect information about their *Students* in connection to *Internships*. Suppose a data set D_1 collected by the first department is described by the attributes *ID*, *Advisor Position*, *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 second data set D_2 collected by the second department is described by the attributes *Student ID*, *Advisor Rank*, *Student Program*, *Hourly Income* and *Intern* and it is stored into a table as the one corresponding to D_2 in Table 1.

A Min Tjoa and J. Trujillo (Eds.): DaWaK 2006, LNCS 4081, pp. 363–373, 2006.

Consider a user, e.g., a university statistician, who wants to draw some inferences about the two departments of interest from the user's perspective, where the representative attributes are *Student SSN*, *Advisor Status*, *Student Status*, *Yearly Income* and *Internship*. For example, the statistician may want to infer a model that can be used to find out whether a student in the statistician's data (D_U in Table 1) has completed an internship or not.

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., *number of doctorate students who did an internship*). However, we notice that the two data sources differ in terms of semantics from the user's perspective. In order to cope with this heterogeneity of semantics, the user must observe that the attributes *ID* in the first data source and *Student ID* in the second data source are similar to the attribute *Student SSN* in the user data; the attributes *Advisor Position* and *Advisor Rank* are similar to the attribute *Advisor Status*; the attributes *Student Level* and *Student Program* are similar to the attribute *Student Status*, etc.

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

	ID	*Adv.Pos.*	*St.Level*	*M.Inc.*	*Intern.*
D_1	34	Associate	M.S.	1530	yes
	49	None	1st Year	600	no
	23	Professor	Ph.D.	1800	no
	SID	*Adv.Rank*	*St.Prog.*	*H.Inc.*	*Intern*
D_2	1	Assistant	Master	14	yes
	2	Professor	Doctoral	17	no
	3	Associate	Undergrad	8	yes
	SSN	*Adv.Status*	*St.Status*	*Y.Inc.*	*Intern*
D_U	475	Assistant	Master	16000	?
	287	Professor	Doctorate	18000	?
	530	Associate	Undergrad	7000	?

To establish the correspondence between values that two similar attributes can take, we need to associate types with attributes and to map the domain of the type of an attribute to the domain of the type of the corresponding attribute (e.g., *Hourly Income* to *Yearly Income* or *Student Level* to *Student Status*). We assume that the type of an attribute can be a standard type such as String, Integer, etc. or it can be given by a simple hierarchical ontology. Figure 1 shows examples of attribute value hierarchies for the attributes *Student Level*, *Student Program*, and *Student Status* in the data sources D_1, D_2 and the user data D_U, respectively. Examples of semantical correspondences in this case could be: $Graduate$ in D_2 is equivalent to $Grad$ in D_U, *1st Year* in D_1 is equivalent to $Freshman$ in D_U, $M.S.$ in D_2 is smaller than (or hierarchically below) $Master$ in D_U, etc.

In this paper, our main focus is on learning classifiers from such semantically heterogeneous data sources. Learning typically requires extracting relevant statistics from data. When the data sources are semantically heterogeneous, because of differences in the levels of abstraction at which data in different data sources are specified relative to the user's perspective, we are presented with the problem of learning classifiers from partially specified data. Previous work [2] has shown how to exploit a set of hierarchically structured ontologies in the form of *isa* hierarchies over attribute values in a single data source to learn classifiers from partially specified data. Against this background, this paper aims to address the problem of *learning concise and accurate classifiers from semantically heterogeneous distributed data sources*.

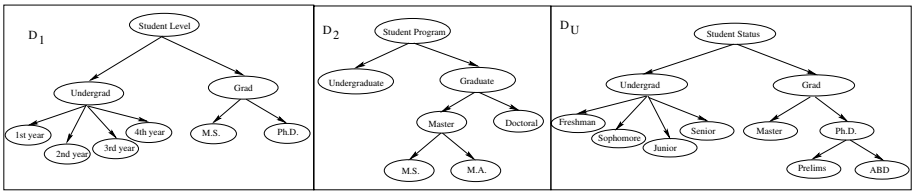

Fig. 1. Hierarchical ontologies associated with the attributes *Student Level*, *Student Program* and *Student Status* that appear in the two data sources of interest D_1 and D_2 and in user data D_U, respectively

The rest of the paper is organized as follows: Section 2 provides a more precise formulation of the problem of learning compact and concise classifiers from semantically heterogeneous distributed data; Section 3 presents a general approach to solving this problem, illustrates its application in the case of Naive Bayes classifiers and presents theoretical guarantees associated with the proposed algorithm; and Section 4 concludes with a summary and discussion.

2 Problem Formulation

2.1 Ontology-Extended Data Sources

Suppose that the data of interest are distributed over the data sources $D_1, \cdots, D_p$, where each data source D_i contains only a fragment of the whole data D. Two common types of data fragmentation are *horizontal fragmentation*, where each data fragment contains a subset of data tuples and *vertical fragmentation*, where each data fragment contains subtuples of data tuples [3].

Let D_i be a distributed data source described by the set of attributes $\{A_1^i, \cdots, A_n^i\}$ and $O_i = \{\Lambda_1^i, \cdots, \Lambda_n^i\}$ a simple ontology associated with this data. The element $\Lambda_j^i \in O_i$ corresponds to the attribute A_j^i and describes the type of that particular attribute. The type of an attribute can be a (possibly restricted) standard type (e.g., Positive Integer or String) or a hierarchical type. A hierarchical type is defined as an ordering of a set of terms [4] (e.g., the values of an attribute). Of special interest to us are tree structured *isa hierarchies* over the values of the attributes that describe a data source, also called *attribute value taxonomies* (see Figure 1).

The schema S_i of a data source D_i is given by the set of attributes $\{A_1^i, \cdots, A_n^i\}$ used to describe the data together with their respective types $\{\Lambda_1^i, \cdots, \Lambda_n^i\}$ described by the ontology O_i. We define an *ontology-extended data source* as a tuple $\mathcal{D}_\rangle = <D_i, S_i, O_i>$, where D_i is the actual data in the data source, S_i is the schema of the data source and O_i is the ontology associated with the data source.

2.2 Complete Data from a User Perspective

Let $<D_1,S_1,O_1>,\cdots, <D_p, S_p, O_p>$ be an ordered set of p ontology-extended data sources and U a user that poses queries against these heterogeneous data sources. A

user perspective is given by a user ontology O_U and a set of interoperation constraints IC that define correspondences between terms in $O_1, \cdots, O_p$ and terms in O_U. The constraints can take one of the forms: $x{:}O_i \equiv y{:}O_U$ (x is semantically *equivalent* to y), $x{:}O_i \leq y{:}O_U$ (x is semantically *below* y), $x{:}O_i \geq y{:}O_U$ (x is semantically *above* y) [4]. The set of constraints specified by the user can be used to (semi-automatically) infer a set of mappings between data source ontologies $O_1, \cdots, O_p$ and a user ontology O_U.

Let $\Gamma = \Gamma(O_U)$ be a cut through the user ontology. If $\Lambda_j^U \in O_U$ is a standard (linear) type, then the cut $\Gamma(\Lambda_j^U)$ through the domain Λ_j^U is the domain itself. However, if Λ_j^U is a hierarchical type, then $\Gamma(\Lambda_j^U)$ defines the level of abstraction at which the user queries are formulated. For example, $\{Undergrad, Master, Ph.D.\}$ is a level of abstraction in the hierarchy associated with the attribute *Student Status* in the user perspective in our example (Figure 1). Any value above this cut implies a higher level of abstraction (e.g., $Grad$), while a value below the cut (e.g., ABD) implies a lower level of abstraction, when used to specify instances. A user level of abstraction Γ determines a level of abstraction $\Gamma_i = \Gamma(O_i)$ in each distributed data source D_i (by applying the corresponding mappings). Let $x = (v(A_1^i), \cdots, v(A_n^i))$ be an instance in D_i. We say that the instance x is:

- *Fully specified* if for all $1 \leq j \leq n$, the value $v(A_j^i)$ is on or below the cut Γ_i. If $v(A_j^i)$ is on the cut Γ_i, we say that $v(A_j^i)$ is an *exactly specified* value; if $v(A_j^i)$ is below the cut Γ_i, we say that $v(A_j^i)$ is an *over-specified* value.
- *Partially specified* if there exist at least one attribute value $v(A_j^i)$ which is above the cut Γ_i. We say that $v(A_j^i)$ is an *under-specified* value.

Given a cut Γ through the user ontology, the available data sources $D_1, \cdots, D_p$ could be seen as a complete virtual data set D, whose instances are specified at the level of abstraction corresponding to the cut Γ. More precisely, D is defined as the multi-set union (i.e., duplicates are allowed) of the distributed instances, appropiately mapped to the user ontology by mapping each attribute value to the corresponding value in the user ontology. Note that the complete data cannot always be constructed in practice (e.g., when the user cut results in under-specified data in the distributed data sources), thus making impossible the application of standard centralized machine learning algorithms. However, under specific assumptions about the distribution of the under-specified data (e.g., all the under-specified values are equally likely), certain statistics about data (e.g., counts of data) can be easily estimated.

2.3 Learning Compact and Accurate Classifiers from Distributed, Ontology-Extended Data Sources

The problem of learning classifiers from data can be summarized as follows [5]: Given a data set D of labeled examples, a hypothesis class H, and a performance criterion P, the learning algorithm L outputs a hypothesis $h \in H$ that optimizes P. In pattern classification applications, h is a classifier (e.g., a Naive Bayes classifiers, a Decision Tree, a Support Vector Machine, etc.). Under appropriate assumptions, the resulting classifier is likely to accurately classify unlabeled instances.

A distributed setting typically imposes a set of constraints Z on the learner that are absent in the centralized setting. In this paper, we assume that the constraints Z prohibit the transfer of raw data from each of the sites to a central location while allowing the learner to obtain certain statistics from the individual sites (e.g., counts of instances that have specified values for some subset of attributes). Thus, the problem of learning compact and accurate classifiers from distributed, semantically heterogeneous data sources can be formulated as follows: Given a collection of ontology-extended data sources $<D_1, S_1, O_1>,\cdots,<D_p, S_p, O_p>$, a user perspective ($O_U$, IC), 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 $h \in H$ that optimizes P using only operations allowed by Z.

We say that an algorithm L_d for learning from distributed, semantically heterogeneous data sets $D_1, \cdots, D_p$ is *exact* relative to its centralized counterpart L if the hypothesis produced by L_d is identical to that obtained by L from the complete data set D obtained by appropriately integrating the data sets $D_1, \cdots, D_p$ according to the user perspective, as defined in the previous section.

3 Sufficient Statistics Based Solution

We want to design algorithms for learning compact and accurate classifiers from distributed, semantically heterogeneous data sources. Our approach is based on a general strategy for transforming algorithms for learning classifiers from data into algorithms for learning classifiers from distributed data [6].

This strategy relies on the decomposition of the learning task into two components [7]: an *information gathering* component, in which the information needed for learning is identified and gathered from the distributed data sources, and a *hypothesis generation* component which uses this information to generate or refine a partially constructed hypothesis. The information gathering component involves a procedure for specifying the information needed for learning as a *query* and a procedure for answering this query from distributed data. The procedure for answering queries from distributed data entails the decomposition of a posed query into sub-queries that the individual data sources can answer, followed by the composition of the partial answers into a final answer to the initial query. If the distributed data sources are also semantically heterogeneous, mappings between the data sources ontologies and a user ontology need to be applied in the process of query answering to reconcile the semantical differences [6] (Figure 2).

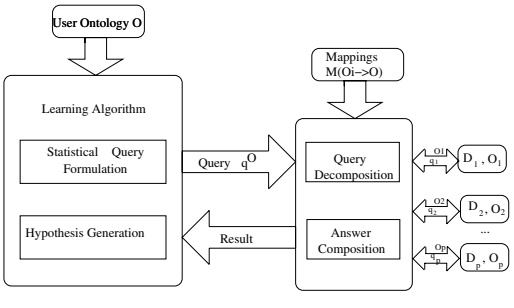

Fig. 2. Learning from semantically heterogeneous data sources

The strategy described can be applied to a large class of learning algorithms (e.g., naive Bayes, decision trees, Bayesian networks, etc.). To illustrate it, we will use Naive

Bayes algorithms as an example. Zhang and Honavar [8] proposed an algorithm (AVT-NBL) for learning compact and accurate Naive Bayes classifiers from a data set in the presence of an associated ontology. In the remaining of this section we identify the information requirements (*sufficient statistics*) of AVT-NBL algorithm, and we show how to transform it into an algorithm for learning compact and accurate Naive Bayes classifiers from distributed, semantically heterogeneous data sources.

3.1 Sufficient Statistics for AVT-NBL

A statistic $s(D)$ is called a *sufficient statistic* for a parameter θ if $s(D)$ captures all the information about the parameter θ contained in the data D [9]. Caragea et al. [6] generalized this notion of a sufficient statistic for a parameter θ to yield the notion of a sufficient statistic $s_L(D)$ for learning a hypothesis h using a learning algorithm L applied to a data set D. Thus, 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.

Consider for example, the Naive Bayes classifier that 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 $c_{MAP}(x)$. Thus, 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 [5] based on relative frequency counts. We denote by $\sigma(v_i|c_j)$ the frequency count of the 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. These frequency counts completely summarize the information needed for constructing a Naive Bayes classifier from D, and thus, they constitute *sufficient statistics* for Naive Bayes learner.

While the sufficient statistics required for constructing a classifier can be computed in one step in some simple cases (e.g., Naive Bayes), in general, this may require interleaved execution of the information gathering and hypothesis generation components of the algorithm over several steps with each step yielding *refinement sufficient statistics* that are used to refine a partially construted classifier. More precisely, $s_L(D, h_i \rightarrow h_{i+1})$ is a sufficient statistic for the refinement of h_i into h_{i+1} if there exists a procedure R that takes h_i and $s_L(D, h_i \rightarrow h_{i+1})$ as inputs and outputs h_{i+1} [3].

We next identify the refinement sufficient statistics for the AVT-NBL algorithm [8]. AVT-NBL efficiently expoits taxonomies defined over values of each attribute in the data set to find a Naive Bayes classifier that optimizes the Conditional Minimum Description Length (CMDL) score [10]. The CMDL score provides a means of trading off the error of the classifier against its complexity. If we denote by

$|D|$ the size of the data set, Γ a cut through the AVT associated with this data, $h = h(\Gamma)$ the Naive Bayes classifier corresponding to the cut Γ, $size(h)$ the number of probabilities used to describe h and $CLL(h|D)$ the conditional log-likelihood of the hypothesis h given the data D, then the $CMDL$ score can be written as $CMDL(h|D) = \left(\frac{\log |D|}{2}\right) size(h) - |D| CLL(h|D)$. Here, $CLL(h|D) = |D| \sum_{i=1}^{|D|} \log p_h(c_i|v_{i1} \cdots v_{in})$, where $p_h(c_i|v_{i1} \cdots v_{in})$ represents the conditional probability assigned to the class $c_i \in C$ associated with the example $x_i = (v_{i1}, \cdots, v_{in})$. Because each attribute is assumed to be independent of the others given the class, we can write $CLL(h|D) = |D| \sum_{i=1}^{|D|} \log \left(\frac{p(c_i) \prod_j p_h(v_{ij}|c_i)}{\sum_{k=1}^{|C|} p(c_k) \prod_j p_h(v_{ij}|c_k)} \right)$.

AVT-NBL starts with a Naive Bayes classifier corresponding to the most abstract cut in the attribute value taxonomy associated with the data (most general classifier) and it iteratively refines the cut by searching in a greedy fashion through the space of possible cuts, 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 3).

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 the differences $|CMDL(\Gamma) - CMDL(\Gamma')|$ approaches zero for all (one-step) refinements Γ' of Γ. The last hypothesis constructed is the output of the AVT-NBL algorithm.

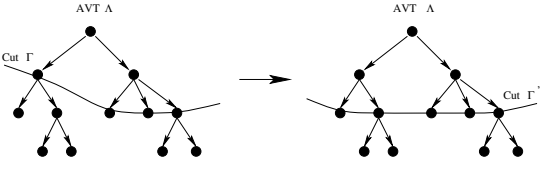

Fig. 3. The refinement of a cut Γ through an attribute value taxonomy Λ

Therefore, the final classifier that the AVT-NBL outputs is obtained from the most general classifier through a sequence of refinement operations. Each refinement operation corresponds to the refinement of the current cut and it is based on the $CMDL$ score. Thus, the sufficient statistics for learning AVT-NBL classifiers can be seen as refinement sufficient statistics, which are identified below.

Let h_i be the current hypothesis corresponding to a cut Γ and $CLDM(\Gamma|D)$ 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}

3.2 Naive Bayes Classifiers from Semantically Heterogeneous Data

Let $<D_1, S_1, O_1>, \cdots, <D_p, S_p, O_p>$ be a set of p ontology-extended data sources and O_U a user ontology. Let Γ be a cut through the user ontology.

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 $D_1, \cdots, D_p$ is similar to the step $i+1$ of the algorithm for learning from a single data set (described above), except that the sufficient statistics are computed from the distributed data sources $D_1, \cdots, D_p$.

Thus, we have reduced the problem of learning Naive Bayes classifiers from distributed, ontology-extended data sources, to the problem of gathering the statistics $s_L(D, h_i \rightarrow h_{i+1})$ from such data sources. Next, we show how to answer statistical queries $q(s_L(D, h_i \rightarrow h_{i+1}))$ that return statistics $s_L(D, h_i \rightarrow h_{i+1})$, from horizontally and vertically fragmented distributed, semantically heterogeneous data sources.

Horizontally Fragmented Data. If the data are horizontally fragmented, the examples are distributed among the data sources of interest. Thus, the user query $q(\sigma(v_i|c_j))$ can be decomposed into the sub-queries $q_1(\sigma(v_i^1|c_j^1)), \cdots, q_p(\sigma(v_i^p|c_j^p))$ corresponding to the distributed data sources $D_1, \cdots, D_p$, where v_i^k and c_j^k are the values in O_k that map to the values v_i and c_j in O_U. Once the queries $q_1(\sigma(v_i^1|c_j^1)), \cdots, q_p(\sigma(v_i^p|c_j^p))$ have been answered, the answer to the initial query can be obtained by adding up the individual answers into a final count $\sigma(v_i|c_j) = \sigma(v_i^1|c_j^1) + \cdots + \sigma(v_i^p|c_j^p)$. Similarly, we compute the counts $\sigma(c_j)$. Once the counts $\sigma(v_i|c_j)$ and $\sigma(c_j)$ have been computed, the Naive Bayes classifier $h' = h(\Gamma')$ corresponding to the cut Γ' can be generated. The next query that needs to be answered is $q(\prod_j p_{h'}(v_{ij}|c_k))$ corresponding to each (virtual) example $x_i = (v_{i1}, \cdots, v_{in})$ (in the complete data set) and each class c_k based on the probabilities that define h'. Because all the attributes of an example are at the same location in the case of the horizontal data fragmentation, each query $q(\prod_j p_{h'}(v_{ij}|c_k))$ is answered by the data source that contains the actual example x_i. When all such queries have been answered, the score $CMDL$ can be computed and thus the hypothesis that will be output at this step can be generated.

If any of the values v_i^k or c_j^k are partially specified in O_k, we "fill in" the partially specified values and increment the count accordingly. Traditional methods for dealing

with missing data, as well as new statistical methods designed specifically for partially specified data can be used to "fill in" partially specified values. In this paper, we assume that the user specifies a distribution over partially specified values or that such a distribution is inferred based on the corresponding specified values in a different data source.

Vertically Fragmented Data. If the data is vertically fragmented, the attributes are distributed among the data sources of interest, but all the values of an attribute are found at the same location. Therefore, a user query $q(\sigma(v_i|c_j))$ can be answered by a particular data source that contains the attribute A_i. However, the user query $q(\prod_j p_h(v_{ij}|c_k))$ is decomposed into sub-queries according to the distributed data sources and the final answer is obtained by multiplying the individual answers.

3.3 Theoretical Analysis

Theorem 1 (Exactness). *The algorithm for learning Naive Bayes classifiers from a set of horizontally (or vertically) fragmented distributed, ontology-extended data sources* $<D_1,S_1,O_1>,\cdots,<D_p,S_p,O_p>$*, from a user perspective* $<O_U, IC>$*, in the presence of the inferred mappings* $\psi_1, \cdots, \psi_p$*, is* exact *with respect to the algorithm for learning Naive Bayes classifiers from the complete data set* D*, obtained (in principle) by integrating the data sources* $D_1, \cdots, D_p$ *according to mappings* $\psi_1, \cdots, \psi_p$*.*

Proof sketch: Because of the information gathering and hypothesis generation decomposition of the the AVT-NBL algorithm, the exactness of the algorithm for learning from distributed, semantically heterogeneous data sources depends on the correctness of the procedures for decomposing a user query q into sub-queries $q_1, \cdots, q_p$ corresponding to the distributed data sources $D_1, \cdots, D_p$ and for composing the individual answers to the queries $q_1, \cdots, q_p$ into a final answer to the query q. More precisely, we need to show that the condition $q(D) = \mathcal{C}(q_1(D_1), \cdots, q_p(D_p))$ (*exactness condition*) is satisfied, where $q(D), q_1(D_1), \cdots, q_p(D_p)$ represent the answers to the queries $q, q_1, \cdots, q_p$, respectively, and $\mathcal{C}$ is a procedure for combining the individual answers.

When data is horizontally fragmented the query $q(\sigma(v_i|c_j))$ is decomposed into sub-queries $q_1(\sigma(v_i^1|c_j^1)), \cdots, q_p(\sigma(v_i^p|c_j^p))$ corresponding to the distributed data sources $D_1, \cdots, D_p$ and the final answer is $\sigma(v_i|c_j)(D_1, \cdots, D_p) = \sigma(v_i^1|c_j^1)(D_1) + \cdots + \sigma(v_i^p|c_j^p)(D_p)$. If we denote by $\sigma(v_i|c_j)(D)$ the answer to the query $q(\sigma(v_i|c_j))$ posed to the complete data set D, we need to show that $\sigma(v_i|c_j)(D_1, \cdots, D_p) = \sigma(v_i|c_j)(D)$. This is obviously true when the data sources $D_1, \cdots, D_p$ are homogeneous because the addition operation is associative. The equality holds also when the data sources are heterogeneous, due to the way we compute the counts (by simulating the construction of the complete data set D). A similar argument can be made for the exactness condition in the case of the query $q(\sigma(c_j))$. Because the answer to the query $q(\prod_j p_h(v_{ij}|c_k))$ is obtained from a single data source and no combination procedure is needed, the exactness condition is trivially satisfied in this case. Similarly we can prove the exactness of the algorithm for leaning from vertically fragmented distributed data, which completes the proof of the exactness theorem.

4 Summary and Discussion

There is an urgent need for algorithms for learning classifiers from distributed, autonomous (and hence inevitably, semantically heterogeneous) data sources in several increasingly data-rich application domains such as bioinformatics, environmental informatics, medical informatics, social informatics, security informatics, among others.

In this paper, we have precisely formulated the problem of learning classifiers from distributed, *ontology-extended data sources*, which make explicit (the typically implicit) ontologies associated with autonomous data sources. User-specified semantic correspondences (mappings between the data source ontologies and the user ontology) are used to answer statistical queries that provide the information needed for learning classifiers, from such data sources. The resulting framework yields algorithms for learning classifiers from distributed, ontology-extended data sources. These algorithms are provably exact relative to their centralized counterparts in the case of the family of learning classifiers for which the information needed for constructing the classifier can be broken down into a set of queries for sufficient statistics that take the form of counts of instances satisfying certain constraints on the values of the attributes. Such classifiers include decision trees, Bayesian network classifiers, classifiers based on a broad class of probabilistic models including generalized linear models, among others. We have illustrated the proposed approach in the case of learning Naive Bayes classifiers from horizontally fragmented distributed, ontology-extended data sources.

There is a large body of literature on distributed learning (See [11] for a survey). However, with the exception of [3], most algorithms for learning classifiers from distributed data do not offer performance guarantees (e.g., exactness) relative to their centralized counterparts. Integration of semantically heterogeneous data has received significant attention in the literature (see [12] for a survey). Most of this work has focused on bridging semantic differences between schemas and ontologies associated with the individual data sources and answering (typically relational) queries from such data sources.

Caragea et al. [6] present an approach to semantic integration of data from multiple sources when data are described in terms of different ontologies and briefly outline some ideas on extending this approach to solve the problem of learning from semantically heterogeneous data. In contrast, this paper precisely formulates and provides a solution to this problem in the important special case where each data source has an AVT ontology associated with it.

The algorithm and the analysis presented in this paper, together with results like those presented in [6] represent important steps towards a problem of significant current interest that cuts across multiple areas of AI (such as informtion integration, machine learning, knowledge representation, etc.).

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001)
2. Zhang, J., Caragea, D., , Honavar, V.: Learning ontology-aware classifiers. In: Proceedings of the Eight International Conference on Discovery Science (DS 2005). (2005) 308–321

3. 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)
4. 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
5. Mitchell, T.: Machine Learning. McGraw Hill (1997)
6. 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)
7. Kearns, M.: Efficient noise-tolerant learning from statistical queries. Journal of the ACM **45** (1998) 983–1006
8. Zhang, J., Honavar, V.: AVT-NBL: An algorithm for learning compact and accurate naive bayes classifiers from attribute value taxonomies and data. In: Proceedings of the Fourth IEEE International Conference on Data Mining, Brighton, UK (2004)
9. Casella, G., Berger, R.: Statistical Inference. Duxbury Press, Belmont, CA (2001)
10. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning **29** (1997)
11. Kargupta, H., Chan, P.: Advances in Distributed and Parallel Knowledge Discovery. AAAI/MIT (2000)
12. Doan, A., Halevy, A.: Semantic Integration Research in the Database Community: A Brief Survey. AI Magazine, Special Issue on Semantic Integration **26** (2005) 83–94