Learning Classifiers from Distributed, Ontology-Extended Data Sources

Doina Caragea, Jun Zhang, Jyotishman Pathak, and Vasant Honavar
Artificial Intelligence Research Laboratory
Department of Computer Science
Iowa State University
226 Atanasoff Hall
Ames, IA 50011, USA
{dcaragea,jzhang,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. given user-supplied semantic correspondences between data source ontologies and the user ontology. 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.

Index Terms

Machine learning, knowledge discovery, semantically heterogeneous data, ontologies, attribute value taxonomies, naive Bayes algorithm.

I. INTRODUCTION

The availability of large amounts of data in many application domains has resulted in great opportunities for data driven knowledge discovery. Inevitably, data collected by different institutions could be semantically heterogeneous, making it difficult to use traditional knowledge discovery techniques. The Semantic Web enterprise [1] aims to support 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. Because users often need to analyze data in different contexts from different perspectives (e.g., in collaborative scientific discovery applications), given a set of distributed data sources and their associated ontologies, there is no single privileged perspective that can serve all users, or for that matter, even a single user, in every context. Effective use of multiple sources of data in a given context requires reconciliation of such semantic differences from a user's point of view. Hence, in this paper we address the problem of learning classifiers from a collection of distributed, semantically heterogeneous data sources viewed from a user perspective, under the assumption that data integration prior to the learning process is not feasible.

We will use a practical example to illustrate the problem that we are addressing. Consider two academic departments that independently collect information about their *Students* in connection to *Internships*. Suppose that the data 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 I.

The data 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 I.

TABLE I

STUDENT DATA COLLECTED BY TWO DEPARTMENTS AND A UNIVERSITY 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	Undergraduate	8	yes
	SSN	*Adv.Status*	*St.Status*	*Y.Inc.*	*Intern*
D_U	475	Assistant	Master	16000	?
	287	Professor	Ph.D.	18000	?
	530	Associate	Undergrad	7000	?

Consider a university statistician (user) who wants to draw some inferences about the two departments of interest from his or her own 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 his or her own data (represented as in the entry corresponding to D_U in Table I) has done 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.

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.

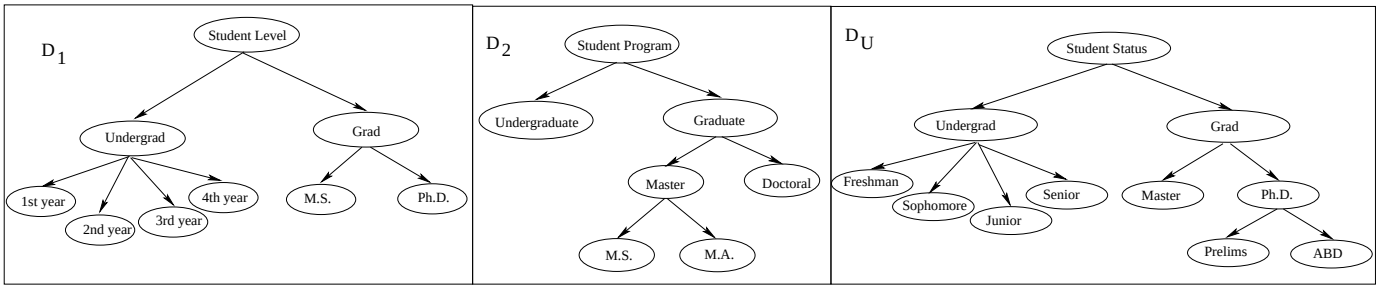

Fig. 1. Hierarchical ontologies associated with the attributes *Student Level*, *Student Program* and *Student Status* that appear in the two data sources of interest D_1 and D_2 and in user data D_U, respectively.

Note that data in different data sources could be described at different levels of abstraction. For instance, the attribute *Student Level* in D_1 is specified in a greater detail (lower level of abstraction) than the corresponding attribute *Student Program* in D_2. Assuming that the desired level of abstraction for the values of the user attribute *Student Status* is $\{Undergrad, Master, Prelim, ABD\}$, then the value *Undergrad* is *over-specified* in D_1, while the values *Prelims* and *ABD* are *under-specified* (or *partially specified*) in both data sources D_1 and D_2. Therefore, learning classifiers

from semantically heterogeneous data sources presents us with the problem of learning classifiers from *partially specified data*.

In this paper, we precisely define the problem introduced informally here and present a sufficient statistics based solution to this problem. The solution can be used to transform a large class of algorithms for learning from data into algorithms for learning from distributed, semantically heterogeneous data. A performance criterion (exactness) for evaluating the resulting algorithms relative to their centralized counterparts is also introduced. We illustrate the proposed approach in the case of learning Naive Bayes classifiers from distributed, ontology-extended data sources and we prove that the resulting algorithm is exact relative to its centralized counterpart.

The rest of the paper is organized as follows: Section 2 precisely formulates the problem addressed. Section 3 presents a general approach to this problem, illustrates the application of this approach to design algorithms for learning Naive Bayes classifiers from semantically heterogeneous data sources and demonstrates the exactness of the resulting algorithms relative to their centralized counterparts. Section 4 concludes with a summary, discussion of related work and ideas for future work.

II. Problem Formulation

A. Ontology-extended data sources

Let D_i be a data set associated with the ith 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 set. 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 (e.g., the values of an attribute) [2]. 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* (AVT). Examples of AVTs are shown in 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, i.e., $S_i = \{A_1^i : \Lambda_1^i, \cdots, A_n^i : \Lambda_n^i\}$. An *ontology-extended data source* is defined as a tuple $\mathcal{D}_i = <D_i, S_i, O_i>$, 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. Obviously, the following condition

needs to be satisfied: $D_i \subseteq \Lambda_1^i \times \cdots \times \Lambda_n^i$, which means that each attribute A_j^i can take values in the set Λ_j^i defined in the ontology O_i.

B. Data Sources 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. After [3], we define a user perspective as consisting of 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), $x{:}O_i \preceq y{:}O_U$ (x is semantically *below* y), $x{:}O_i \succeq y{:}O_U$ (x is semantically *above* y) [2].

We say that the ontologies $O_1,\cdots,O_p$ are integrable according to the user ontology O_U in the presence of the interoperation constraints IC if there exist p partial injective mappings $\psi_1,\cdots,\psi_p$ from $O_1,\cdots,O_p$, respectively, to O_U with the following two properties [2], [3]:

(a) For all $x,y \in O_i$, if $x \preceq y$ in O_i then $\psi_i(x) \preceq \psi_i(y)$ in O_U (order preservation property);

(b) For all $x \in O_i$ and $y \in O_U$, if $(x : O_i \text{ op } y : O_U) \in IC$, then $\psi_i(x)$ op y in the ontology O_U (interoperation constraints preservation property).

A set of candidate mappings that are consistent with the interoperation constraints can be automatically inferred. A user can inspect the set of candidate mappings and accept, reject or modify them [3].

Given $<D_1,S_1,O_1>,\cdots,<D_p,S_p,O_p>$, a set of p distributed, ontology-extended data sources, O_U, a user ontology and $\psi_1,\cdots,\psi_p$, a set of inter-ontology mappings, the data sets $D_1,\cdots,D_p$ specify a virtual data set D, as it will be explained below.

Let $\Gamma = \Gamma(O_U) = \{\Gamma(O_1),\cdots,\Gamma(O_p)$ be a cut through the user ontology. Note that if $\Lambda_j^U \in O_U$ is a standard type (e.g., Integer), 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. 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). We say that an instance $x_i \in D_i$ is *partially specified* if there exist at least one attribute value $v(A_j^i)$ in x_i which is *partially specified*, i.e., $v(A_j^i)$ is above the level of abstraction Γ_i in Λ_i.

Thus, if the user ontology describes data at a lower level of abstraction than one or more data source ontologies, the resulting data set D is *partially* specified from the user's perspective. In

order to deal with partially specified values, additional assumptions about the distribution of the partially specified values need to be made by the user. For example, in some cases it may be reasonable to assume a uniform distribution over the partially specified values. In other cases, the user may assume that all the data sources (or a subset of data sources) come from the same distribution and, hence, the distribution infered from a data source where all the values are fully specified can be assumed also for a data source that contains partially specified values.

Once the distributional assumptions concerning partially specified data are specified, a *virtual data set* D can be constructed by generating from each partially specified instance, several *fractionally weighted, fully specified* instances based on the observed distribution of values of the corresponding attribute(s) at the desired level of abstraction in O_U. In other words, the distribution of attribute values in the resulting fractionally weighted instances are identical to the the corresponding distributions in the fully specified instances under the user-specified distributional assumptions.

Two common types of data fragmentation are of interest in the distributed setting [4]:

1) **Horizontal fragmentation**: D is obtained by the multi-set union (i.e., duplicates are allowed) of $D_1 \ldots D_p$ viewed from the user perspective (after appropriate mappings are applied). Thus, $D = \psi(D_1) \cup \cdots \cup \psi(D_p)$, where $\psi(D_i) = \{\psi(x_i) | x_i \in D_i\}$ and $\psi(x_i) = w_i \cdot (\psi(v(A_1^i)), \cdots, \psi(A_n^i))$ for each $x_i = (v(A_1^i), \cdots, v(A_n^i))$ in D_i. The weight w_i is 1 if all the values in $\psi(x_i)$ are specified, otherwise it is obtained based on the distribution assumed.
2) **Vertical fragmentation**: Individual data sources store values for (possibly overlapping) subsets of the attributes used to describe the data. To keep things simple, we assume that there is a unique index that can be used to easily assemble the instances of D from the corresponding instance fragments stored in $D_1 \cdots D_p$ (after applying the appropriate mappings as in the horizontally fragmented case).

Note that an ontology-extended data source $< D_i, O_i, S_i >$ can have data specifed at a lower or higher level of abstraction with respect to the associated ontology, which can also result in partially specified data. A detailed description of how we can deal with such cases can be found in [5]. Here, we assume only partial specification that appears as a result of applying mappings, as this is specific to the distributed case.

C. Learning classifiers from distributed, ontology-extended data sources

The problem of learning from data can be summarized as follows [6]: Given a data set D, 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.). The data D typically consists of a set of training examples. The goal of learning is to produce a hypothesis that optimizes the performance criterion of minimizing some function of the classification error (on the training data) and the complexity of the hypothesis. Under appropriate assumptions, this is likely to result in a classifier that assigns correct labels to 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 classifiers from ontology-extended 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) which implies a set of mappings $\psi_1, \cdots, \psi_p$, a set of assumptions $\mathcal{A}$ with respect to the distributions of the partially specified values resulted as effect of applying the mappings, 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 $D_1, \cdots, D_p$, under the assumptions $\mathcal{A}$, 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 set of mappings $\psi_1, \cdots, \psi_p$ and the assumptions $\mathcal{A}$, as described in the previous section.

III. Sufficient Statistics Based Approach

Our approach to the problem of learning classifiers from from distributed, ontology-extended data sources is based on a general strategy for transforming algorithms for learning classifiers from data into algorithms for learning classifiers from distributed data [4].

This strategy relies on the decomposition of the learning task into two components: 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 in order to reconcile the semantical differences [3] (Figure 2). The exactness of the solution depends on the correctness of the procedure for query decomposition and answer composition.

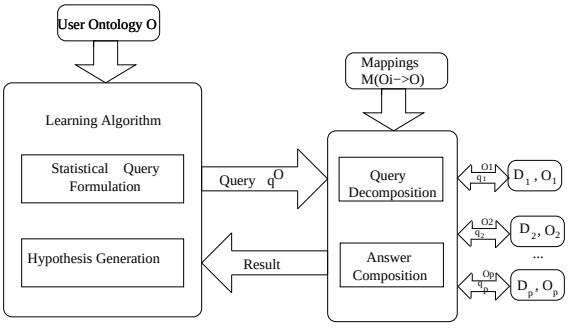

Fig. 2. Learning from distributed, semantically heterogeneous data sources.

The strategy described can be applied to a large class of learning algorithms (e.g., Naive Bayes, Decision Trees, Support Vector Machines, etc.) wherein the information needed for constructing the classifier from data can be obtained using a suitable set of statistical queries from the data. We illustrate the application of this approach in the case of learning Naive Bayes classifiers from distributed, semantically heterogeneous data.

A. Sufficient statistics for Naive Bayes classifiers

According to the classical statistical theory [7], 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. For example, sample mean is a sufficient statistic for the mean of a Gaussian distribution.

This notion of a sufficient statistic for a parameter θ can be generalized 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 [4]. 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.

We want to identify sufficient statistics for Naive Bayes classifier, a simple and yet effective classifier that has performance comparable to the performance of other more sophisticated classifiers [6]. 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 [6] 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.

As noted above (problem formulation), learning classifiers (and in particular Naive Bayes classifiers) from semantically heterogeneous data sources presents us with the problem of partially specified data. AVT-NBL [5] is an example of an algorithm for learning Naive Bayes classifiers that can handle partially specified data. In addition, AVT-NBL can efficiently exploit attribute value taxonomies as opposed to the traditional Naive Bayes algorithm, feature that makes it appropriate for our setting.

We have seen that the sufficient statistics for the Naive Bayes algorithm can be computed in one step. As opposed to this, the sufficient statistics for AVT-NBL are computed by interleaving the information gathering and hypothesis generation components several times. The sufficient statistics computed at each step are called *refinement sufficient statistics* as they are used to refine a partially construted hypothesis. 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} [4].

We show how refinement sufficient statistics can be used to transform AVT-NBL into an algorithm for learning Naive Bayes classifiers from distributed, semantically heterogeneous data.

AVT-NBL finds a Naive Bayes classifier that optimizes a performance criterion, called Conditional Minimum Description Length (CMDL) score [8], defined as a tradeoff between the accuracy and the complexity of the classifier. 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 $h_0 = h(\Gamma_0)$ 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 3). Let $h(\Gamma')$ be the Naive

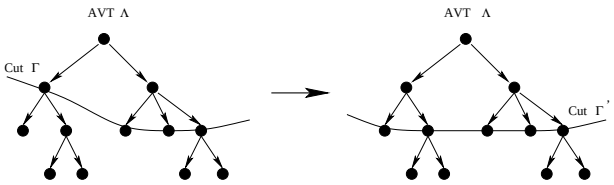

Fig. 3. The refinement of a cut Γ through an attribute value taxonomy Λ.

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}

B. Naive Bayes classifiers from semantically heterogeneous data

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 information gathering and hypothesis generation decomposition of the AVT-NBL algorithm, 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.

1) Horizontally fragmented data: If the data are horizontally fragmented, the instances 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 deal with them as described in Section 2.2, except that we do not explicitly construct the transformed instances (according to the distribution assumed by the user), but implicitly use them for the computation of the (fractional) counts.

Note that the set of class conditional counts $\sigma(v_i|c_j)$, corresponding to the values v_i of an attribute A_k, can be represented as a tree (whose structure is given by the associated AVT) and can be efficiently computed using the approach described in [5].

2) Vertically fragmented data: In the case of vertical data fragmentation, the attributes are distributed among the data sources of interest, but all the values of an attribute are found at the same location. We assume that each location contains the class attribute. To answer the user query

$q(\sigma(v_i|c_j))$, this query is sent to the particular data source D_k that contains the attribute A_i after being mapped to the query $q_k(\sigma(v_i^k|c_j^k))$, where the values v_i^k and c_j^k in O_k are the correspondents of the values v_i and c_j, respectively, in O_U. The answer to the query $q_k(\sigma(v_i^k|c_j^k))$ is the final answer to the user query $q(\sigma(v_i|c_j))$. Because the class attribute is present at each location, the query $q(\sigma(c_j))$ can be answered by any data source D_k after being appropriately mapped to the ontology O_k. Because the attributes are distributed at different locations, the user query $q(\prod_j p_h(v_{ij}|c_k))$ is decomposed into the sub-queries $q_1(\prod_{j_1} p_h(v_{ij_1}^1|c_k^1)), \cdots, q_p(\prod_{j_p} p_h(v_{ij_1}^p|c_k^p))$, where each j_s ($s = \overline{1,p}$) belongs to set of indices corresponding to the attributes that are located at the site k and the values $v_{ij_s}^s$, c_k^s are the correspondents in O_s of the values v_{ij}, c_k in O_U. Once these queries are answered by the distributed data sources, the answer to the initial user query is obtained by multiplying the partial answers into a final answer $\prod_j p_h(v_{ij}|c_k)) = \prod_{j_1} p_h(v_{ij_1}^1|c_k^1) \times \cdots \times \prod_{j_p} p_h(v_{ij_p}^p|c_k^p)$. We deal with partially specified values as in the case of horizontal data fragmentation.

C. Theoretical Analysis

Theorem [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 mappings $\psi_1, \cdots, \psi_p$, under a set of user-specified distributional assumptions $\mathcal{A}$ regarding partially specified data, is exact with respect to the algorithm for learning Naive Bayes classifiers from the complete virtual fully specified data set D, constructed by integrating the data sources $D_1, \cdots, D_p$ according to the mappings $\psi_1, \cdots, \psi_p$ and assumptions $\mathcal{A}$.

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 in the case of semantically heterogeneous data because the relevant counts are computed under identical distributional assumptions concerning partially specified data (or equivalently, from the same fully specified virtual 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. Thus, we showed that the exactness condition holds for all queries that are posed in the process of computing the sufficient statistics needed to learn Naive Bayes classifiers from horizontally fragmented distributed, semantically heterogeneous data sources. This completes the proof of the exactness theorem for the horizontally fragmented case.

A similar argument can be made for the vertically fragmented case.

IV. Summary, Discussion and Future Work

A. Summary

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 and vertically fragmented distributed, ontology-extended data sources.

B. Discussion

There is a large body of literature on distributed learning (See [9] for a survey). However, with the exception of [4], 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 [10] for a survey). Most of this work has focused on bridging semantic differences between ontologies associated with the individual data sources and answering (typically relational) queries from such data sources [2], [3].

McClean *et al.* [11], [12] present an approach to answering aggregate queries formulated in a global ontology, from statistical databases. However, they do not address the problem of answering statistical queries from relational data from a user's point of view. Kearns [1998] describe the use of a statistics oracle to extend sample complexity results derived in the *probably approximately correct* (PAC) learning framework to learning scenarios in which the data is corrupted by noisy attribute values and class labels. In previous work [5], we formulated and solved the problem of learning Naive Bayes classifiers from data given an ontology in the form of a set of attribute values taxonomies (one AVT per attribute), in a setting in which the values of some of the attributes are partially specified relative to the corresponding AVT.

In contrast, this paper precisely formulates and solves the problem of learning classifiers from semantically heterogeneous data sources in the important special case where each data source has associated with it, an ontology that takes the form of a set of AVT (with one AVT per attribute per data source). The approach described here builds on our previous work on a sufficient-statistics based general strategy for learning classifiers from (semantically homogeneous) distributed data [4], and on learning Naive Bayes classifiers from (semantically homogeneous) partially specified data [5] to develop for the first time, a provably sound approach to learning classifiers from

semantically heterogeneous distributed data.

C. Future Work

Some promising directions for further work include:

- Application of the general framework described in this paper to obtain algorithms for learning decision trees, Bayesian networks, neural networks, support vector machines and other types of classifiers, and more generally, predictive models including in particular, multi-relational models from semantically heterogeneous ontology-extended data sources.
- Development of sound approaches to answering statistical queries from ontology-extended data sources under a broad range of access, bandwidth, and processing constraints associated with the data sources, including methods for resource-bounded approximations of answers to statistical queries
- Large scale application of the resulting algorithms to data-driven classifier construction problems that arise in bioinformatics and related applications.

ACKNOWLEDGMENTS

This research was supported in part by grants from the National Science Foundation (NSF IIS 0219699) and the National Institutes of Health (GM 066387).

REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, May 2001.

[2] P. Bonatti, Y. Deng, and V. Subrahmanian, "An ontology-extended relational algebra," in *Proceedings of the IEEE Conference on Information Integration and Reuse*. IEEE Press, 2003, pp. 192–199.

[3] D. Caragea, J. Pathak, and V. Honavar, "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.

[4] D. Caragea, A. Silvescu, and V. Honavar, "A framework for learning from distributed data using sufficient statistics and its application to learning decision trees," *International Journal of Hybrid Intelligent Systems*, vol. 1, no. 2, 2004.

[5] J. Zhang and V. Honavar, "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.

[6] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[7] G. Casella and R. Berger, *Statistical Inference*. Belmont, CA: Duxbury Press, 2001.

[8] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, 1997.

[9] H. Kargupta and P. Chan, *Advances in Distributed and Parallel Knowledge Discovery*. AAAI/MIT, 2000.

[10] A. Levy, "Logic-based techniques in data integration," in *Logic-based artificial intelligence*. Kluwer Academic Publishers, 2000, pp. 575–595.

[11] S. McClean, R. Páircéir, B. Scotney, and K. Greer, "A negotiation agent for distributed heterogeneous statistical databases," *SSDBM 2002*, pp. 207–216, 2002.

[12] S. McClean, B. Scotney, and K. Greer, "A scalable approach to integrating heterogeneous aggregate views of distributed databases," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, pp. 232–235, 2003.

[13] M. Kearns, "Efficient noise-tolerant learning from statistical queries," *Journal of the ACM*, vol. 45, no. 6, pp. 983–1006, 1998. [Online]. Available: citeseer.nj.nec.com/kearns93efficient.html