Learning Link-Based Naïve Bayes Classifiers from Ontology-Extended Distributed Data

Cornelia Caragea[1], Doina Caragea[2], and Vasant Honavar[1]

[1] Computer Science Department, Iowa State University
[2] Computer and Information Sciences, Kansas State University
{cornelia,honavar}@cs.iastate.edu,
dcaragea@ksu.edu

Abstract. We address the problem of learning predictive models from multiple large, distributed, autonomous, and hence almost invariably semantically disparate, relational data sources from a user's point of view. We show under fairly general assumptions, how to exploit data sources annotated with relevant meta data in building predictive models (e.g., classifiers) from a collection of distributed relational data sources, without the need for a centralized data warehouse, while offering strong guarantees of *exactness* of the learned classifiers relative to their centralized relational learning counterparts. We demonstrate an application of the proposed approach in the case of learning link-based Naïve Bayes classifiers and present results of experiments on a text classification task that demonstrate the feasibility of the proposed approach.

1 Introduction

Recent advances in sensors, digital storage, computing, and communications technologies have led to a proliferation of autonomously operated, distributed data repositories in virtually every area of human endeavor. Many groups have developed approaches for querying semantically disparate sources [1,2,3,4], for discovering semantic correspondences between ontologies [5,6], and for learning from autonomous, semantically heterogeneous data [7]. One approach to learning from semantically disparate data sources is to first integrate the data from various sources into a warehouse based on semantics-preserving mappings between the data sources and a global integrated view, and then execute a standard learning algorithm on the resulting centralized, semantically homogeneous data. Given the autonomous nature of the data sources on the Web, and the diverse purposes for which the data are gathered, it is unlikely that a unique global view of the data that serves the needs of different users or communities of users under all scenarios exists. Moreover, in many application scenarios, it may be impossible to gather the data from different sources into a centralized warehouse because of restrictions on direct access to the data. This calls for approaches to learning from semantically disparate data that do not rely on direct access to the data but instead can work with results of statistical queries against

R. Meersman, T. Dillon, P. Herrero (Eds.): OTM 2009, Part II, LNCS 5871, pp. 1139–1146, 2009.

an integrated view. We present a principled approach to the problem of learning classifiers from a collection of semantically disparate *relational* data sources in such a setting. We use link-based Naïve Bayes classifiers as an example to illustrate this approach. We show, under fairly general assumptions, that our approach is guaranteed to yield classifiers that are identical to those obtained from a centralized, integrated data warehouse constructed from the collection of semantically disparate relational data sources and associated ontologies and mappings. Experimental results using our implementation of link-based Naïve Bayes classifiers [8,9] for constructing text classifiers from text repositories based on related but disparate ontologies demonstrate the feasibility of the proposed approach.

2 Learning Classifiers from Semantically Heterogeneous Relational Data

2.1 Ontology-Extended Relational Data Sources and User Views

An ontology O associated with a relational data source D is given by a content ontology that describes the semantics of the content of the data (the values and relations between values that the attributes can take in D)[1]. Of particular interest are ontologies that specify *hierarchical* relations among values of the attributes. *Isa* relations induce *attribute value hierarchies* (AVHs) over values of the corresponding attributes. Thus, an ontology O consists of a set of AVHs $\{\mathcal{T}_1, \cdots, \mathcal{T}_l\}$, w.r.t. the *isa* relation. A *cut* (or *level of abstraction*) through an AVH induces a partition of the set of leaves in that hierarchy. A *global cut* through an ontology consists of a set of cuts, one for each constituent AVH.

Figures 1(a) and 1(b) show two AVHs over the values of two attributes `Article.`*Topic* and `Article.`*Words*, respectively, corresponding to a concept `Article` of a bibliographic domain. The set of values of `Article.`*Topic* consists of {Artificial Intelligence (AI), Data Mining (DM), Machine Learning (ML), Natural Language Processing (NLP), Neural Networks (NN), Genetic Algorithms (GA), Case-Based (CB), Probabilistic Methods (PM), Theory (T), Reinforcement Learning (RL)}. $\{DM, ML, NLP\}$ represents a cut Γ through the AVH in 1(a). $\{DM, NN, GA, CB, PM, T, RL, NLP\}$ is a refinement of Γ.

Definition: An *ontology-extended relational data source* (OERDS) [10] is defined as a tuple $\mathcal{D} = \{S, D, O\}$, where S represents the relational data source schema (concepts, their attributes, and the relations between concepts), D is an instantiation of S, and O represents the data source ontology.

A *mapping* ψ from a user ontology O_U to a data source ontology O_D (defining the semantics of two different views of the same domain) establishes semantic correspondences between the values of the attributes in O_U and the values of attributes in O_D. Examples of such semantic correspondences are equality, $x = y$

[1] In a more general setting, an *ontology O* contains also a *structure ontology* that describes the semantics of the elements of a schema S (concepts, their attributes, and the relations between concepts), in addition to the *content ontology.*

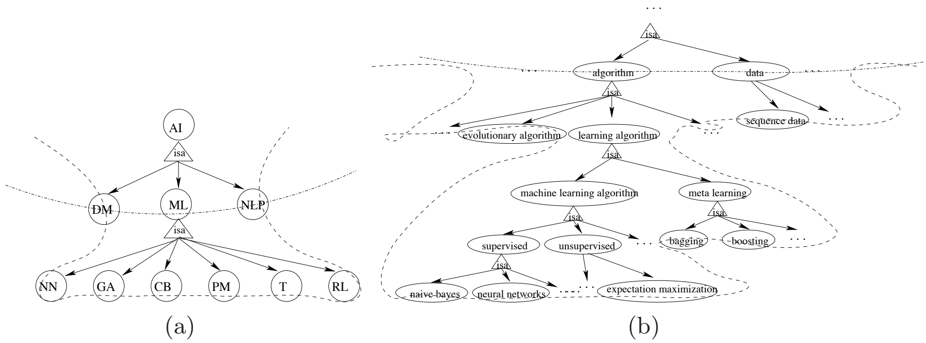

Fig. 1. Two Attribute Value Hierarchies (AVHs) over the values of attributes `Article.`*Topic* (a) and `Article.`*Words* (b), respectively, corresponding to the bibliographic domain. The dash curves represent different levels of abstraction.

(i.e., x and y are *equivalent*), and inclusion $x < y$ (i.e., *y subsumes x*, or y is *more general* than x) [11]. A subset of semantic correspondences between two AVHs corresponding to two ontologies O_U and O_D, $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively, over the values of `Article.`*Topic* is $\{DM_{\mathcal{T}_1} = DataMining_{\mathcal{T}_2}$, $NN_{\mathcal{T}_1} < MachineLearning_{\mathcal{T}_2}$, $AI_{\mathcal{T}_1} > DataMining_{\mathcal{T}_2}\}$.

Definition: Let $\mathcal{D}_1$=$\{S_1, D_1, O_1\}$,$\cdots$,$\mathcal{D}_p$=$\{S_p, D_p, O_p\}$ be a set of OERDSs. A user ontology O_U, together with a set of mappings $\{\psi_k | k = 1, \cdots, p\}$ from O_U to the data source ontologies $O_1, \cdots, O_p$ define a *user view* [10,7].

The user view *implicitly* specifies a user level of abstraction, corresponding to the leaf nodes of the hierarchies in O_U. The mappings ψ_k can be established manually or semi-automatically (e.g., using existing approaches to learning mappings between ontologies [12]).

2.2 Learning Classifiers from OERDSs

We assume the existence of: (1) A collection of several related OERDSs $\mathcal{D}_1$=$\{S_1, D_1, O_1\}$,$\cdots$,$\mathcal{D}_p$=$\{S_p, D_p, O_p\}$ for which the schemas and the ontologies are made *explicit* and the instances in the data sources are labeled according to some criterion of interest to a user (e.g., topic categories); (2) A user view, consisting of a user ontology O_U and a set of mappings ψ_k that relate O_U to $O_1, \cdots, O_p$; (3) A hypothesis class H (e.g., Bayesian classifiers) defined over an *instance space*; (4) A performance criterion P (e.g., accuracy on a classification task).

Under the above assumptions, learning classifiers from a collection of semantically heterogeneous OERDSs can be formulated as follows: *the task of a learner L is to output a hypothesis* $h \in H$ *that optimizes P, via the mappings* $\{\psi_k\}$.

In this setting, the statistical query answering component of the algorithm poses a *statistical query* against the user view; decomposes the query into subqueries and translates them into queries that can be answered by the individual data sources (based on the mappings from the user ontology to the data source

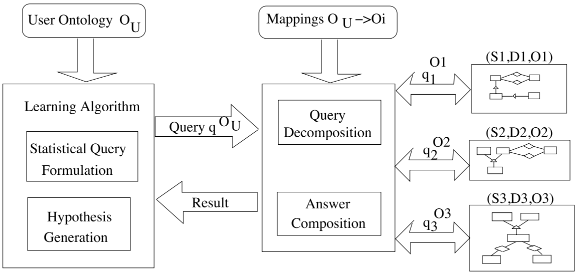

Fig. 2. Learning classifiers from OERDSs [10]

ontologies); and assembles the answer to the original query from the answers returned by the individual data sources (Figure 2). Once a classifier has been learned, it can be used to classify data that is at the disposal of the user.

3 Learning Link-Based Classifiers from OERDs

We now proceed to describe our algorithm for learning classifiers from a collection of semantically heterogeneous OERDSs. We adapt the link-based iterative algorithm introduced by Lu and Getoor [8] to learning classifiers from OERDSs (see [13] for more details on the algorithm adaptation).

Learning link-based Naïve Bayes classifiers reduces to estimating the probabilities $P(c_j)$, $P(v_i|c_j)$, and $P(u_i|c_j)$, for all classes $c_j \in \mathbf{C}$, for all object attribute values $v_i \in \mathcal{V}(OA(x_i))$ and for all link description values $u_i \in \mathcal{V}(LD_l(x_i))$ using standard methods [9] (see [8] for an explanation of the link description).

We denote by $\sigma(v_i|c_j)$ the frequency count of the value $v_i \in \mathcal{V}(OA(x_i))$, given the class c_j; by $\sigma(u_i|c_j)$ the frequency count of the value $u_i \in \mathcal{V}(LD_l(x_i))$, given the class c_j; and by $\sigma(c_j)$ the frequency count of the class c_j, in the user view. The algorithm for learning a link-based Naïve Bayes classifier from a set of related OERDSs works as follows:

1. Select a global user cut Γ through the user ontology (AVHs). In particular, the user cut corresponds to the set of primitive values (i.e., leaves in AVHs).
2. Apply the mappings ψ_k to find a cut Γ_k, corresponding to the user cut Γ, in each OERDS $\mathcal{D}_k$.
3. Formulate statistical queries asking for the frequency counts $\sigma(v_i|c_j)$, $\sigma(u_i|c_j)$, and $\sigma(c_j)$, using terms in the user cut Γ.
4. Translate these queries into queries expressed in the ontology of each OERDS $\mathcal{D}_k$, using terms in the cut Γ_k, and compute the local counts $\sigma_k(v_i|c_j)$, $\sigma_k(u_i|c_j)$, and $\sigma_k(c_j)$ from each OERDS $\mathcal{D}_k$.
5. Send the local counts to the user and add them up to compute the global frequency counts $\sigma(v_i|c_j)$, $\sigma(u_i|c_j)$, and $\sigma(c_j)$.
6. Generate the link-based Naïve Bayes h_Γ corresponding to the cut Γ based on the global frequency counts.

3.1 Exactness

Definition: An algorithm $\mathcal{L}_{distributed}$ for learning from OERDSs $\mathcal{D}_1, \cdots, \mathcal{D}_p$, via the mappings $\{\psi_k\}$, is *exact* wrt its centralized counterpart $\mathcal{L}_{centralized}$, if the hypothesis produced by $\mathcal{L}_{distributed}$ is identical to that produced by $\mathcal{L}_{centralized}$ from the data warehouse D that is constructed by integrating the data sources $D_1, \cdots, D_p$, according to the user view, via the same mappings $\{\psi_k\}$.

The *exactness* criterion defined above assumes that it is possible, in principle, to create an integrated data warehouse in the centralized setting. In practice, the data sources $D_1, \cdots, D_p$ might impose access constraints on the user. These constraints might prohibit the user from retrieving instance data from some of the data sources (e.g., due to restrictions on the queries that can be answered by the data source, bandwidth limitations, or privacy considerations), while allowing retrieval of answers to statistical queries against the data.

Note that the algorithm for learning a link-based Naïve Bayes classifier from OERDSs using statistical queries is *exact* relative to the link-based Naïve Bayes classifier obtained by executing the standard algorithm on the data warehouse $\mathcal{D}$ obtained by integrating the set of OERDSs $\mathcal{D}_1, \cdots, \mathcal{D}_p$ (using the same set of mappings $\{\psi_k\}$). This follows from the observation that $\sigma(v_i|c_j) = \sum_{i=1}^{k} \sigma_k(v_i|c_j) = \sigma_{\mathcal{D}}(v_i|c_j)$, $\sigma(u_i|c_j) = \sum_{i=1}^{k} \sigma_k(u_i|c_j) = \sigma_{\mathcal{D}}(u_i|c_j)$, $\sigma(c_j) = \sum_{i=1}^{k} \sigma_k(c_j) = \sigma_{\mathcal{D}}(c_j)$, when there is no overlap between the distributed sources. Note that dealing with duplication of instances between any two data sources requires establishing correspondences between individual instances [14].

4 Experiments and Results

We evaluated our approach to learning classifiers from a set of semantically disparate relational data sources on a subset extracted from the Cora data set [15]. The filtering procedure of the Cora is described in [13]. We associate AVHs with both attributes `Article`.*Words* and `Article`.*Topic* (see [13]).

Note that due to the unavailability of data sources that are already annotated with meta data, we performed experiments only on the Cora data set. To simulate the distributed setting, we randomly partitioned the Cora data set into two subsets, such that the class distribution in each subset is similar to the class distribution in the entire dataset. In our experiments, we used one-to-one, manually-defined mappings between the user and the data sources ontologies[2].

Futhermore, four cuts, or *levels of abstraction*, through the user AVH corresponding to the `Article`.*Words* were considered. These cuts are obtained as follows. In each hierarchy the most abstract level, i.e. the terms corresponding to the children of the root form **Cut 1**. The most detailed level, i.e. the terms corresponding to the leaves of the trees form the **Leaf Cut**. **Cut 2** is obtained by

[2] There are several approaches to inferring mappings between ontologies from available information [12]. Our focus here is on how to exploit ontologies and mappings, and not the problem of coming up with the mappings.

Table 1. The classification results on the task of classifying papers into one of the three categories: *DM*, *ML*, and *NLP* for all four levels of abstraction considered: **Cut 1**, **Cut 2**, **Cut 3**, **Leaf Cut**

Level of Abstraction	Accuracy	Precision	Recall	F-Measure
Cut 1	0.86	0.80	0.47	0.51
Cut 2	0.86	0.83	0.46	0.51
Cut 3	**0.89**	**0.86**	**0.62**	**0.69**
Leaf Cut	**0.89**	0.84	0.61	0.68

replacing one node from **Cut 1** by its children. **Cut 3** is obtained by replacing a subset of leaf nodes by their parent node.

We learned classifiers using terms on different cuts (*levels of abstraction*) in the ontologies. Assume that a user is interested in classifying computer science research articles into one of the three classes: *DM*, *ML* and *NLP* and also that the user provides a level of abstraction corresponding to his or her understanding of the domain, i.e. a level of abstraction in the AVH corresponding to the `Article`.*Words* attribute.

The classification results for this task, for all four levels of abstraction, **Cut 1**, **Cut 2**, **Cut 3**, and **Leaf Cut**, are shown in Table 1. The performance measures of interest were estimated by averaging the performance of the classifier on the five runs of a cross-validation experiment. As can be seen from the table, classifiers trained at different levels of abstraction differ in their performance on the test data. Moving from a coarser to a finer level of abstraction does not necessarily improve the performance of the classifier because there may not be enough data to accurately estimate the classifier parameters. Similarly, moving from a finer to a coarser level of abstraction does not necessarily improve the performance since there may not be enough terms to discriminate between the classes. **Cut 3** yields the best performance among the four levels considered, although it is an abstraction of the **Leaf Cut**.

Now assume that another user is interested in predicting whether the topic of a research article is *NN*. This requires finding a cut through the user AVH corresponding to `Article`.*Topic* that contains the term *NN* and then performing the mappings between the user ontology and the data source ontologies.

Figure 3(a) shows the Receiver Operating Characteristic (ROC) curves on this binary classification task using the same four levels of abstraction as above. As can be seen from the figure, for any choice of the FPR, as we go from a coarser to a finer level of abstraction, the link-based Naïve Bayes classifier offers a higher TPR (Recall). The performance improvement is quite striking from **Cut 1** to **Cut 2**. However, the difference in performance between **Cut 3** and **Leaf Cut** is rather small. Unlike the previous task, on this task the ROC curve for the **Leaf Cut** outperforms the ROC curves corresponding to the other three cuts. This can be explained by the fact that the number of parameters that need to be estimated is smaller for this second task.

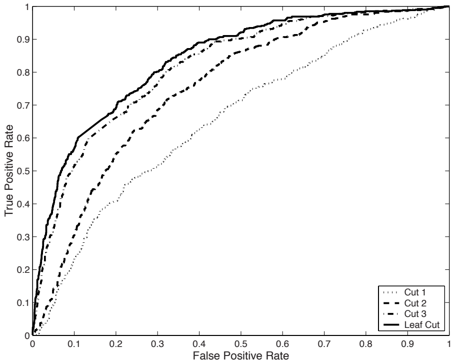

Fig. 3. Comparison of the ROC curves of the link-based classifier on the task of predicting whether a research paper is *NN* for all four levels of abstraction considered in this study: **Cut 1**, **Cut 2**, **Cut 3**, and **Leaf Cut**

5 Summary and Discussion

We have described a general strategy for learning link-based Naïve Bayes classifiers [8] from a collection of semantically disparate relational data sources. The proposed approach exploits mappings between a user ontology and data source ontologies to gather the necessary statistics for learning the classifiers. The resulting algorithms for learning link-based classifiers from semantically disparate relational data sources can be shown to be *provably exact* relative to their centralized counterparts under fairly general assumptions. The algorithm assumes a pre-specified *level of abstraction* defined by the user-supplied global cut through the user ontology. Our experiments have shown that the choice of the level of abstraction can impact the performance of the classifier.

The problem of learning classifiers from a semantically homogeneous relational database has received much attention in the recent machine learning literature [16,17]. There is a large body of literature on learning predictive models from distributed data (see [18,19] for surveys). Of particular interest in our setting is the work of Caragea et al [7] that introduced a general strategy for transforming a broad class of standard learning algorithms that assume in memory access to a dataset into algorithms that interact with the data source(s) only through statistical queries or procedures that can be executed on the data sources. A basic strategy for coping with semantically disparate data was outlined in [7]. However, each of these works assumed that data are stored in a *flat* table.

Some directions for future research include: exploring the effect of using different ontologies and mappings, the effect of degree of incompleteness of mappings, the effects of errors in mappings, the use of automated approaches to establish mappings between ontologies [12], coping with partially specified data [20] that inevitably results by integrating a collection of OERDSs via mappings (when different data sources might specify data at different levels of abstraction), etc.

References

1. Levy, A.: Logic-based techniques in data integration. In: Logic-based artificial intelligence, pp. 575–595. Kluwer Academic Publishers, Dordrecht (2000)
2. Noy, N.F.: Semantic Integration: A Survey Of Ontology-Based Approaches. SIGMOD Record, Special Issue on Semantic Integration 33 (2004)
3. Doan, A., Halevy, A.: Semantic Integration Research in the Database Community: A Brief Survey. AI Magazine 26, 83–94 (2005)
4. Calvanese, D., De Giacomo, G., Lenzerini, M., Vardi, M.Y.: View-based query processing: On the relationship between rewriting, answering and losslessness. In: Eiter, T., Libkin, L. (eds.) ICDT 2005. LNCS, vol. 3363, pp. 321–336. Springer, Heidelberg (2005)
5. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: The state of the art. In: Proceedings of Semantic Interoperability and Integration, Dagstuhl, Germany (2005)
6. Noy, N., Stuckenschmidt, H.: Ontology Alignment: An annotated Bibliography. In: Semantic Interoperability and Integration. Dagstuhl Seminar Proceedings, vol. 04391 (2005)
7. Caragea, D., Zhang, J., Bao, J., Pathak, J., Honavar, V.: Algorithms and software for collaborative discovery from autonomous, semantically heterogeneous information sources. In: Proceedings of ICALT, Singapore. LNCS, pp. 13–44 (2005)
8. Lu, Q., Getoor, L.: Link-based classification. In: Proceedings of the International Conference on Machine Learning, ICML (2003)
9. Mitchell, T.: Machine Learning. McGraw Hill, New York (1997)
10. Caragea, D., Bao, J., Honavar, V.: Learning relational bayesian classifiers on the semantic web. In: Proceedings of the IJCAI 2007 SWeCKa Workshop, India (2007)
11. Rajan, S., Punera, K., Ghosh, J.: A maximum likelihood framework for integrating taxonomies. In: Proceedings of AAAI, Pittsburgh, Pennsylvania, pp. 856–861 (2005)
12. Doan, A., Madhavan, J., Dhamankar, R., Domingos, P., Halevy, A.: Learning to match ontologies on the semantic web. VLDB Journal (2003)
13. Caragea, C., Caragea, D., Honavar, V.: Learning link-based classifiers from ontology-extended textual data. In: Proceedings of ICTAI 2009, Newark, New Jersey, USA (2009)
14. Parag, Domingos, P.: Multi-relational record linkage. In: Proceedings of the KDD-2004 Workshop on Multi-Relational Data Mining, Seattle, CA. ACM Press, New York (2004)
15. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Automating the contruction of internet portals with machine learning. Information Retrieval Journal 3, 127–163 (2000)
16. Getoor, L., Friedman, N., Koller, D., Taskar, B.: Learning probabilistic models of relational structure. Journal of Machine Learning Research 3, 679–707 (2002)
17. Neville, J., Jensen, D., Gallagher, B.: Simple estimators for relational bayesian classifiers. In: Proceedings of the 3rd IEEE ICDM 2003 (2003)
18. Kargupta, H., Chan, P.: Advances in Distributed and Parallel Knowledge Discovery. AAAI/MIT (2000)
19. Caragea, D., Honavar, V.: Learning classifiers from distributed data sources. Encyclopedia of Database Technologies and Applications (2008)
20. Zhang, J., Honavar, V.: Learning decision tree classifiers from attribute-value taxonomies and partially specified data. In: Fawcett, T., Mishra, N. (eds.) Proceedings of ICML, Washington, DC, pp. 880–887 (2003)