Learning Link-Based Classifiers from Ontology-Extended Textual Data

Cornelia Caragea
Computer Science Department
Iowa State University
cornelia@cs.iastate.edu

Doina Caragea
Computer and Information Sciences
Kansas State University
dcaragea@ksu.edu

Vasant Honavar
Computer Science Department
Iowa State University
honavar@cs.iastate.edu

Abstract

Real-world data mining applications call for effective strategies for learning predictive models from richly structured relational data. In this paper, we address the problem of learning classifiers from structured relational data that are annotated with relevant meta data. Specifically, we show how to learn classifiers at different levels of abstraction in a relational setting, where the structured relational data are organized in an abstraction hierarchy that describes the semantics of the content of the data. We show how to cope with some of the challenges presented by partial specification *in the case of structured data, that unavoidably results from choosing a particular level of abstraction. Our solution to partial specification is based on a statistical method, called* shrinkage. *We present results of experiments in the case of learning link-based Naïve Bayes classifiers on a text classification task that (i) demonstrate that the choice of the level of abstraction can impact the performance of the resulting link-based classifiers and (ii) examine the effect of partially specified data.*

1. Introduction

Advances in sensors, digital storage, computing and communications technologies have led to an exponential increase in the amount of on-line richly structured, relational data. Problems such as classifying web pages, filtering e-mail, annotating images, etc., have become very important. Machine learning algorithms [11], [1] offer some of the most cost effective approaches to building predictive models (e.g., classifiers) in a broad range of applications in data mining.

Although the problem of learning classifiers from relational data sources has received much attention in the machine learning literature [3], [12], [15], there has been limited exploration of techniques that use structured relational data sources that are annotated with relevant meta data. In this paper, we address the problem of learning classifiers from such data.

Representational commitments, i.e., the choice of features or attributes that are used to describe the data presented to a learner, and the level of detail at which they describe the data, can have a major impact on the difficulty of learning, and the accuracy, complexity, and comprehensibility of the learned predictive model [17]. The representation has to be rich enough to capture distinctions that are relevant from the standpoint of learning, but not so rich as to make the task of learning infeasible due to overfitting.

Hence, we present an approach to learning classifiers at different *levels of abstraction* (or detail) when the data (attributes and classes) are organized in abstraction hierarchies. We adapt the link-based iterative classification algorithm introduced by Lu and Getoor [6] and use it in conjunction with Naïve Bayes classifiers to illustrate our approach.

Furthermore, we show how to cope with *partially specified data* that inevitably result from choosing a particular level of abstraction. Zhang et al. [18], [19] have previously addressed the problem of partially specified data in cases where data instances are described by nominal attributes. In this study, we deal with partial specification in the case of structured data, where, for example, a multinomial model is assumed as the underlying model that generated the data instances. We use a statistical approach, called *shrinkage*, to cope with partially specified data.

We evaluated our approach to learning classifiers from structured relational data annotated with relevant meta data on the *Cora* data set, a standard relational benchmark data set of research articles and their citations [9] for which we manually constructed an abstraction hierarchy from the words in the data set. The task was to classify research articles based on their topics. The results of our experiments show that more abstract levels can yield better performing link-based classifiers, due to more robust estimates of model parameters (smaller number of parameters that need to be estimated from data). Moreover, the results show that making use of partially specified data can improve classification performance.

The rest of the paper is organized as follows: In Section 2, we introduce the framework necessary for learning classifiers from structured relational data annotated with relevant meta data. We present our approach to learning classifiers from such data in the case of link-based Naïve Bayes classifiers [6] in Section 3. In Section 4, we discuss some of the challenges presented by partially specified data in our setting. Section 5 provides experimental results on a relational data set from the text categorization applications. Section 6 concludes with summary, a brief discussion of related work, and an outline of some directions for further research.

2. Ontology-Extended Structured Relational Data

A *schema* $\mathcal{S}$ of a structured relational data source describes a set of *concepts* $\mathcal{X} = \{X_1, \cdots, X_t\}$, and the *relations* between them, $\mathcal{R}(X_i, X_j)$. An instance of a concept X_i is a structured object, e.g. a string, a document, or an image. These instances can be described by *features* such as a set of attributes, a histogram of words, or features from spatially contiguous elements of an image. An attribute A of a concept X_i, denoted by $X_i.A$ takes values in a set $\mathcal{V}(X_i.A)$. A relation $R(X_i, X_j)$ corresponds to a set defined as $x_i.R = \{x_j \in X_j$ s.t. x_j is related to x_i through $R\}$, where x_i denotes an instance of the concept X_i. A tuple (x_i, x_j) is an instance of the relation R. A *data set* $\mathcal{D}$ that specifies a set of instances and the relations between them is an *instantiation* $\mathcal{I}(\mathcal{S})$ of a schema $\mathcal{S}$. It can be represented as an *instantiation graph* in which nodes denote instances and edges denote relations between instances [3].

Figure 1a shows a simple schema of a bibliographic domain which consists of the `Article` concept and the `Cites` relation. Figure 1b shows a sample instantiation graph corresponding to the relational schema in Figure 1a. The nodes x_i in the graph represent research articles (i.e., `Article` instances, $x_i \in$ `Article`) and the edges (x_i, x_j) represent the "citation" relation between articles (i.e., $(x_i, x_j) \in$ `Cites`). Note that each instance can have a variable number of related instances and thus, a variable number of *features* [12].

Assuming that we model a research article using the histogram of word occurrences, the concept `Article` is described by two attributes: `Article.`*Words* that denotes the word histogram of the article, and `Article.`*Topic*, a categorical attribute that denotes the topic of the article.

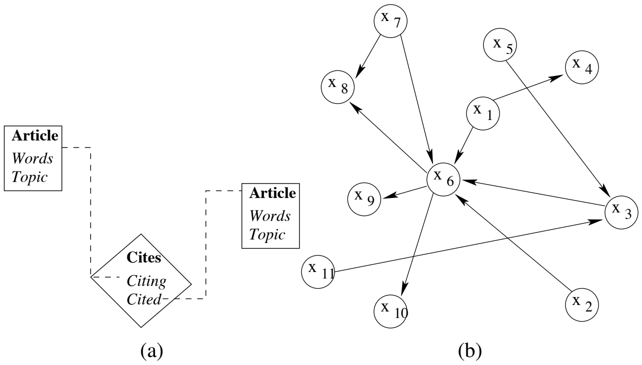


Figure 1: (a) A simple schema corresponding to a bibliographic domain (the figure is adapted from Getoor et al. [3]). (b) A sample instantiation graph corresponding to the schema in (a). The nodes x_i represent `Article` instances, i.e., $x_i \in$ `Article`, while the edges (x_i, x_j) represent `Cites` instances, i.e., $(x_i, x_j) \in$ `Cites`.

An ontology $\mathcal{O}$ associated with a structured relational data source $\mathcal{D}$ is given by a *content ontology* that describes the semantics of the content of the data (e.g., the values and relations between values that `Article.`*Words* can take in $\mathcal{D}$)[1]. Of particular interest are ontologies that specify *hierarchical* relations among values of attributes. *Isa* relations induce *abstraction hierarchies* (AHs) over the values of attributes.

Definition 1 (Abstraction Hierarchy) *An abstraction hierarchy $\mathcal{T}$ associated with an attribute $X_i.A$ is a rooted tree such that: (1) The tree $\mathcal{T}$ has exactly $n = |\mathcal{V}(X_i.A)|$ nodes such that the leaves correspond to the most specific values of $X_i.A$, and the internal nodes correspond to abstractions over the most specific values of $X_i.A$ (i.e., more abstract values of $X_i.A$); in particular, the root of $\mathcal{T}$ corresponds to the most abstract value of $X_i.A$; and (2) The edges of $\mathcal{T}$ represent partial order relations $\prec$ (e.g., isa relations) between their corresponding nodes.*

Definition 2 (m-Cut) *An m-cut (or level of abstraction) γ_m through the abstraction hierarchy $\mathcal{T}$ is a subset of m nodes of $\mathcal{T}$ satisfying the following properties: (1) For any leaf a_i, either $a_i \in \gamma_m$ or a_i is a descendant of a node $a_j \in \gamma_m$; and (2) For any two nodes $a_k, a_l \in \gamma_m$, a_k is neither a descendant nor an ancestor of a_l. The set of abstractions $\mathcal{A}$ at any given m-cut γ_m forms a partition of the set of leaves.*

Figures 2a and 2b show two fragments of AHs over the values of the attributes `Article.`*Topic* and

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

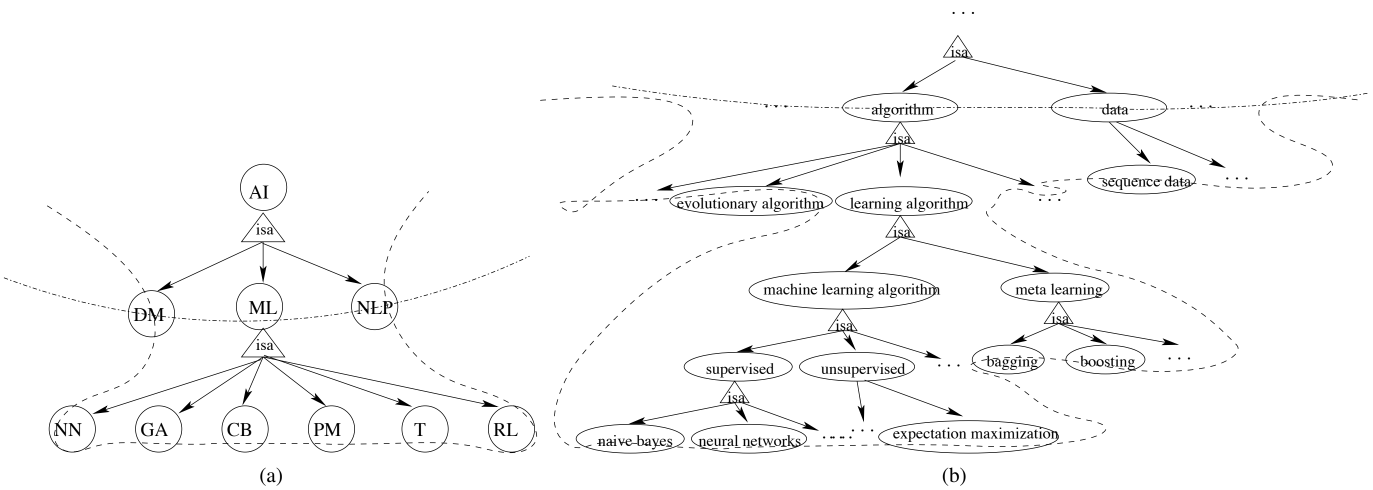


Figure 2: Two fragments of abstraction hierarchies (AHs) over the values of attributes `Article`.*Topic* and `Article`.*Words* corresponding to a bibliographic domain are shown in (a) and (b), respectively. The dash curves represent different cuts. The set {*DM, ML, NLP*} represents a cut or *level of abstraction* γ_3 in (a). {*DM, NN, GA, CB, PM, T, RL, NLP*} is a finer cut γ_8.

`Article`.*Words*, respectively, corresponding to the bibliographic domain. The set $\mathcal{V}(\texttt{Article}.Topic)$ 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)}. The subset of nodes $\{DM, ML, NLP\}$ represents a 3-cut γ_3 through the AH in 2a. The subset $\{DM, NN, GA, CB, PM, T, RL, NLP\}$ is a finer cut γ_8.

An ontology $\mathcal{O}$ associated with a structured relational data source $\mathcal{D}$ consists of a set of AHs $\{\mathcal{T}_1, \cdots, \mathcal{T}_l\}$, corresponding to the set of attributes, w.r.t. the *isa* relation. A global cut Γ through $\mathcal{O}$ consists of a set of cuts, one for each constituent AH, e.g., $\Gamma = \{\gamma_{Words}, \gamma_{Topic}\}$.

The *isa* relations between the nodes in an AH $\mathcal{T}_i$ associated with an attribute $X_i.A$ specify semantic relationships between the values of the attribute (more specific and more abstract values). Examples of such semantic relationships are equality, $x = y$, meaning that x and y are *equivalent* (or *synonyms*), and inclusion $x < y$, meaning that y *subsumes* x, or y is *more general* than x [13]. A subset of inclusion relationships between the values of `Article`.*Topic* is {$NN < ML$, $AI > ML$}.

Definition 3 (Ontology-Extended Structured Relational Data Source) *An ontology-extended structured relational data source (OESRDS) is defined as a tuple* $\{\mathcal{S}, \mathcal{D}, \mathcal{O}\}$*, where* $\mathcal{S}$ *represents the structured relational data source schema,* $\mathcal{D}$ *is an instantiation of* $\mathcal{S}$*, and* $\mathcal{O}$ *represents the data source ontology [2].*

3. Learning Link-Based Classifiers from OESRDSs

We now proceed to describe an algorithm for learning classifiers from OESRDSs. We adapt the link-based iterative classification algorithm introduced by Lu and Getoor [6] to the problem of learning classifiers from OESRDSs. We apply the resulting classifiers on the bibliographic domain where the task is to classify research articles based on their topics.

3.1. Link-based iterative classification

To label an instance, the iterative classification algorithm learns and exploits the distribution of links in the instantiation graph, in addition to the information available in the instance itself. This approach to learning from relational data is potentially more powerful than methods (e.g., influence propagation over relations [16]) that assume that related instances have similar labels.

An `Article` instance x_i is represented using the attribute `Article`.*Words*, denoted by $OA(x_i)$, and the link distribution of x_i, denoted by $LD(x_i)$ (to exploit the link patterns in classifying x_i). The object attribute $OA(x_i)$ holds x_i's word frequency counts. The link distribution $LD(x_i)$ holds the topic frequency counts computed from the set of objects that are linked to x_i in four different ways as follows:

- $InLink(x_i) = \{x_j \text{ s.t. } (x_j, x_i) \in \texttt{Cites}\}$,
- $OutLink(x_i) = \{x_j \text{ s.t. } (x_i, x_j) \in \texttt{Cites}\}$,

- $CoInLink(x_i) = \{x_j$ s.t. $x_j \neq x_i$ and $\exists x_k$: $(x_k, x_i) \in \texttt{Cites}$ and $(x_k, x_j) \in \texttt{Cites}\}$,
- $CoOutLink(x_i) = \{x_j$ s.t. $x_j \neq x_i$ and $\exists x_k$: $(x_i, x_k) \in \texttt{Cites}$ and $(x_j, x_k) \in \texttt{Cites}\}$.

The iterative classification algorithm consists of two steps, bootstrap and iteration [6]:

Step 1: Using only the object attributes $OA(x_i)$, assign an initial topic to each article x_i in the test set.

Step 2: Using the object attributes $OA(x_i)$ and the link description $LD(x_i)$, iteratively assign a topic to each article x_i in the test set, until a termination criterion is met (e.g., either there are no changes to the topic assignments of articles or a certain number of iterations is reached). That is, for each article x_i:

1. Encode x_i using $OA(x_i)$ and $LD(x_i)$, based on the current assignments of linked articles;
2. Compute: $\hat{c}(x_i) =$

$$\arg\max_{c_j \in \mathbf{C}} P(c_j|OA(x_i)) \prod_{l \in \{In, Out, CI, CO\}} P(c_j|LD_l(x_i))$$

We use a Multinomial Naïve Bayes classifier [8]. The Naïve Bayes classifier makes the assumption that the attributes of each instance are conditionally independent given the class. Using Bayes rule and the independence assumption, the above probabilities can be replaced by:

$$P(c_j|OA(x_i)) = P(c_j) \prod_{v_i \in \mathcal{V}(OA(x_i))} P(v_i|c_j)$$

$$P(c_j|LD_l(x_i)) = P(c_j) \prod_{u_i \in \mathcal{V}(LD_l(x_i))} P(u_i|c_j)$$

Although this assumption may be violated in practice, empirical results show that Naïve Bayes classifier is competitive with state-of-the-art methods (including those that are computationally far more expensive than Naïve Bayes) on the document classification task [11, 12].

3.2. Learning Link-Based Naïve Bayes Classifiers from OESRDSs

We note that the task of 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 class labels $c_j \in \mathbf{C}$, for all object attribute values $v_i \in \mathcal{V}(OA(x_i))$ and for all link description values $u_i \in \mathcal{V}(LD_l(x_i))$. These probabilities can be estimated from data using standard methods [11]. The resulting estimates constitute *sufficient statistics* for the parameters that specify a link-based Naïve Bayes classifier.

The task of learning link-based Naïve Bayes classifiers from an OESRDS $\{\mathcal{S}, \mathcal{D}, \mathcal{O}\}$ at a given level of abstraction Γ essentially entails estimating the relevant probabilities from the corresponding OESRDS.

We denote by $\sigma(v_i|c_j)$ the frequency counts of the value $v_i \in \mathcal{V}(OA(x_i))$, given the class label c_j; by $\sigma(u_i|c_j)$ the frequency counts of the value $u_i \in \mathcal{V}(LD_l(x_i))$, given the class label c_j; and by $\sigma(c_j)$ the frequency counts of the class label c_j, for a particular choice of a level of abstraction Γ in $\mathcal{O}$. The algorithm for learning a link-based Naïve Bayes classifier from an OESRDS works as follows:

1. Formulate statistical queries asking for the frequency counts $\sigma(v_i|c_j)$, $\sigma(u_i|c_j)$, and $\sigma(c_j)$, using the terms on the global cut Γ (γ_{Words} and γ_{Topic}).
2. Generate the link-based Naïve Bayes h_Γ corresponding to the cut Γ based on the computed frequency counts.

Choosing a level of abstraction Γ in $\mathcal{O}$ can result in data that are only *partially specified*. This can arise from *partially specified values* of an attribute[2]. In the next section, we define the partially specified values and provide a solution to the problem of dealing with partially specified data based on a statistical approach, called *shrinkage* [14], [4], [10].

4. Coping with Partially Specified Data

Definition 4 (Partially Specified Value): *An attribute value $v_i \in \mathcal{V}(X_i.A)$ in an AH $\mathcal{T}$ is partially specified (or under-specified) w.r.t. an attribute value $v_j \in \mathcal{V}(X_i.A)$ in the same AH if $v_i > v_j$; v_i is over-specified w.r.t. v_j if $v_i < v_j$; v_i is fully-specified w.r.t. v_j if $v_i = v_j$ [19].*

For example, given the AH in Figure 2a over the values of the attribute `Article`.*Topic*, the value ML is under-specified w.r.t. NN, since a machine learning article may be a neural network article, a reinforcement learning article, etc., but over-specified w.r.t. AI because each machine learning article is an artificial intelligence article. Furthermore, ML is fully-specified w.r.t. $MachineLearning$ (not shown in the figure).

The problem of learning from partially specified data (when only the attributes are partially specified) has been addressed by Zhang et al. [18, 19] in the setting where data instances are described by nominal attributes. This approach exploits the observation that the problem of learning from partially specified data is a natural generalization of the problem of learning from data in the presence of missing attribute values [19]. Hence, it is possible to adapt statistical approaches for dealing with missing data [5] to deal with partially specified data *under a variety of assumptions*,

[2] Partially specified data can also arise from partially specified schemas, i.e., when schema concepts are partially specified.

(e.g., the distribution of an under-specified attribute value is similar to that in another data source where the corresponding attribute is fully specified).

Learning from data instances (e.g., documents) where a multinomial model is assumed as the underlying generative model presents further complications: The "same" attribute can be over-specified in one place in the document and under-specified in another place in the document. To see this, consider the following paragraph taken from John C. Mallery's article "Semantic Content Analysis: A New Methodology for the RELATUS Natural Language Environment" [7]:

"*Semantic content analysis differs from traditional computerized content analysis because it operates on the referentially integrated meaning representation of a text instead of a linear string of words. Rather than assessing the thematic orientation of texts based on the frequencies of word occurrences, this new methodology examines and interprets explicit knowledge representations of texts. [· · ·] Beyond semantic content analysis, lexical classification expands the referential performance because it provides a basic inference mechanism to extend indexation, semantically disambiguate word senses, and provide criteria for further deliberation in reference. [· · ·] The immediate political-analytic application of lexical classification is semantic content analysis.*"

In this paragraph, the term *semantic content analysis* is over-specified w.r.t. *lexical classification* which in turn is over-specified w.r.t. *methodology*; the term *this new methodology* refers to the *semantic content analysis*.

The fact that a term can be at the same time over-specified and under-specified even in the same document (Figure 3), complicates the problem of dealing with partially specified data. Our solution to this problem is based on a statistical method, called *shrinkage* [14], [4], [10], that provides better estimates of a model parameters when the data is organized in an abstraction hierarchy (as defined in Section 2). Hence, we compute the counts for a term T_i in a document D as a summation (or *cumulative term frequency counts*) of the following three types of counts:

1. the term T_i frequency counts, i.e., the number of T_i occurrences in the document D;

2. the sum of term frequency counts coming from all its descendants T_k in the hierarchy, i.e., the term frequency counts of each node term T_k in the tree rooted at T_i (the term T_i is under-specified wrt any of its descendants);

3. a percentage term frequency counts coming from all its ancestors in the hierarchy, i.e., the percentage term frequency counts of each node term T_j in the tree on the path from T_i to the root of the tree. Thus, the frequency counts of term T_j are distributed among all its descendants proportionally to their frequency counts (the term T_i is over-specified wrt any of its ancestors).

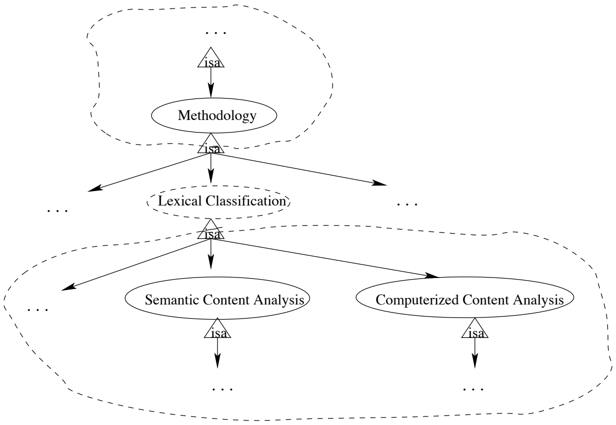


Figure 3: A fragment of an AH associated with the attribute *methodology*. The term *lexical classification* is at the same time under-specified (wrt *semantic content analysis*) and over-specified (wrt *methodology*) in the same document. Its cumulative frequency counts are computed from its own frequency counts, term frequency counts coming from all its descendants, and a percentage term frequency counts coming from all its ancestors. The participating nodes are dashed.

5. Experiments and Results

5.1. Experiments

The goal of the experiments is to explore the feasibility of our approach to learning link-based Naïve Bayes classifiers from ontology-extended structured relational data sources. We investigated: (i) the effect of learning classifiers at different levels of abstraction; (ii) the effect of partially specified data on the performance of our classifiers by comparing two approaches: using only term frequency counts and using cumulative term frequency counts.

We evaluated our approach on a subset of the Cora data set [9], that is a standard benchmark data set of research articles and their citations (which can be modeled as relations among the articles). The article topics are organized in a hierarchy with 73 leaves. We considered only the articles in Cora that are found on the Web and have topics in the topic hierarchy shown in Figure 2a, and that cite or are cited by at least one other article, so that there are no isolated nodes in our instantiation graph. Filtering the Cora data using these criteria yields a data set of 2469 articles and 8297 citations. We associate *abstraction hierarchies* (AHs) over the values of both attributes of the concept `Article`, i.e. `Article`.*Words* and `Article`.*Topic*.

Topic	NumArticles
Neural Networks (NN)	610
Genetic Algorithms (GA)	342
Case-Based (CB)	242
Probabilistic Methods (PM)	339
Theory (T)	325
Reinforcement Learning (ReL)	214
Data Mining (DM)	167
Natural Language Processing (NLP)	236

Table 1: Article topics along with their numbers in our subset of the Cora set.

The `Article`.*Topic* hierarchy is a subtree of the Cora topic hierarchy and contains only 8 leaves out of 73. These leaves along with their article numbers are shown in Table 1. The first six topics can be grouped into the more general term Machine Learning. Machine Learning, Data Mining, Natural Language Processing can be grouped into the more general term Artificial Intelligence (Figure 2a).

We designed the `Article`.*Words* abstraction hierarchy as follows: from the titles and abstracts of our collection of articles, we first removed punctuation and words that contain numbers, and then performed stemming and removal of stop words and words that occur less than 10 times in the whole collection. From the remaining distinct terms we extracted 234 terms and manually organized them in an abstraction hierarchy using definitions in *Wikipedia* (available at http://en.wikipedia.org). Each node in the hierarchy corresponds to one of the 234 terms, and its associated term is more general w.r.t. any term from its descendants and more specific w.r.t. any term from its ancestors in the hierarchy.

In our experiments, we use four cuts, or *levels of abstraction*, through the abstraction hierarchy corresponding to the `Article`.*Words* attribute. These cuts are as follows:

- the most abstract level, i.e. the set of nodes corresponding to the children of the root form the first cut, denoted by **Cut 1**
- the second cut was obtained by replacing one node (randomly chosen) from **Cut 1** by its children; the resulting cut is denoted by **Cut 2**
- the most detailed level, i.e. the set of nodes corresponding to the leaves of the trees form the third cut, denoted by **Leaf Cut**
- a subset of nodes from the **Leaf Cut** was replaced by their parent, denoted by **Cut 3**.

5.2. Results

The effect of learning classifiers at different levels of abstraction. In our first set of experiments, we investigated the effect of learning classifiers at different *levels of abstraction*. We consider two tasks. In the first task suppose that we are interested in classifying computer science research articles into one of the three classes: *Data Mining*, *Machine Learning* and *Natural Language Processing*. Assume that we are given a cut in the AH corresponding to the `Article`.*Words* attribute. Sufficient statistics (corresponding to terms on the cut) are gathered so that the classifier can be trained.

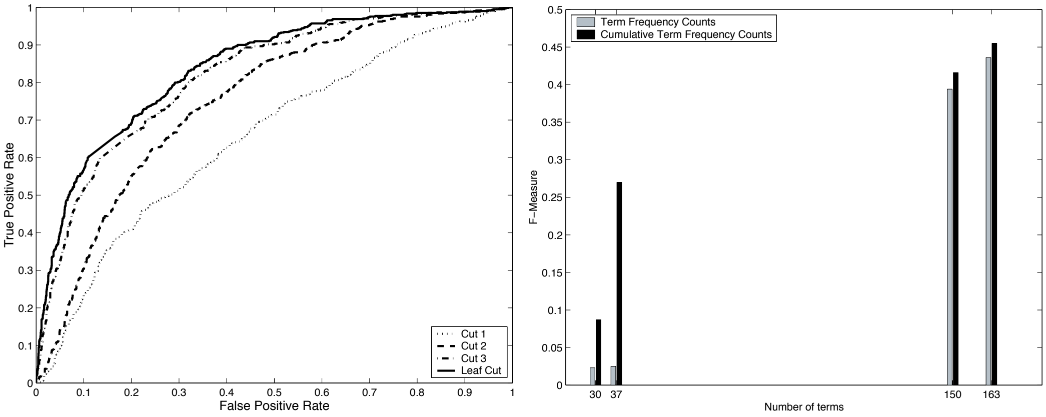
The classification results for this task, for all four levels of abstraction, **Cut 1**, **Cut 2**, **Cut 3**, and **Leaf Cut**, are shown in Table 2. The performance measures of interest were estimated by averaging the performance of the classifier on the five runs of a cross-validation experiment. 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 more general to a more specific 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 more specific to a more general level of abstraction does not necessarily improve the performance since there may not be enough terms on the cut to discriminate between classes. As can be seen in Table 2, **Cut 3** yields the best performance among the four levels considered, although it is an abstraction of the **Leaf Cut**. This suggests the possibility of variants of the algorithm considered here that automatically search for an optimal level of abstraction using methods similar to those proposed in [18], [19].

In the second task suppose that we are interested in predicting whether the topic of a research article is *Neural Networks*. This requires finding a cut through the AH corresponding to the attribute `Article`.*Topic* that contains the term *Neural Networks*. The articles labeled with the term *Neural Networks* represent positive instances, while the rest represent negative instances.

Figure 4a shows the Receiver Operating Characteristic (ROC) curves on this binary classification task using the same four levels of abstraction as above. The ROC curves show the tradeoff between true positive and false positive predictions over their entire range of possible values. As can be seen from the figure, for any choice of the False Positive Rate, as we go from a coarser to a finer level of abstraction, the link-based Naïve Bayes classifier offers a higher True Positive Rate (Recall). The performance improvement is quite striking from **Cut 1** to **Cut 2**. However, the difference in performance between **Cut 3** and the **Leaf Cut** is rather small (and eventually levels off). Unlike the first task where the classifier trained based on **Cut 3** outperforms those trained based on the other cuts, in the second task the classifier trained based on the **Leaf Cut** outperforms the others. This can be explained by the fact that the number of parameters that need to be estimated is much

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

Table 2: The classification results on the task of classifying articles into one of the three categories: *Data Mining*, *Machine Learning*, and *Natural Language Processing* for all four levels of abstraction considered: **Cut 1**, **Cut 2**, **Cut 3**, **Leaf Cut**.



(a)

(b)

Figure 4: a) Comparison of the ROC curves of the link-based Naïve Bayes classifiers on the task of predicting whether a research article is *Neural Networks* for all four levels of abstraction considered in this study: **Cut 1**, **Cut 2**, **Cut 3**, and **Leaf Cut**. b) Comparison of the average F-Measure under two scenarious: the counts for each term are simply term frequency counts (shown in gray) and the counts for each term are cumulative term frequency counts (shown in black). The numbers on the x axes represent the number of terms on a particular cut.

smaller for this second task. As the number of parameters that need to be estimated increases, more and more data is required to obtain good estimates.

The effect of partially specified data on the performance of classifiers. In our second set of experiments, we investigated the effect of *partially specified data* on the performance of our classifiers.

Figure 4b compares the average F-Measure under two scenarios. In the first scenario, the counts for each term on the cut are simply the term frequency counts conditioned on the class attribute, i.e. the number of term occurrences in the collection of articles given the class attribute. In the second scenario, the counts for each term on the cut are the cumulative term frequency counts conditioned on the class attribute (see Section §4 for more details). As can be seen from the figure, taking into account the effect of partially specified data through the means of cumulative term frequency counts improves the average F-Measure for all four levels of abstraction considered in this study on the task of predicting whether the topic of an article is *Neural Networks*. We obtained similar results on the task of classifying articles into one of the three categories: *Data Mining*, *Machine Learning*, and *Natural Language Processing* (data not shown).

6. Summary and Discussion

We have described an algorithm for learning link-based Naïve Bayes classifiers from ontology-extended structured relational data sources. We have evaluated the resulting classifiers on a text categorization task for several choices of levels of abstraction. The results of our experiments show that more abstract levels can yield better performing classifiers. We have also addressed some of the unique challenges presented by partial specification of data which is unavoidable on text data.

The problem of learning classifiers from relational data sources has received much attention in the machine learning literature [3], [12], [15]. In contrast to these methods, we have presented in this paper an algorithm for learning predictive models from relational data sources which is annotated with relevant meta data. Zhang et al. [18], [19] proposed an approach to learning classifiers from partially specified data over nominal attributes when data are stored in a *fla*t table. McCallum et al. [10] have used a well-known statistical approach, called *shrinkage*, in a hierarchy of classes to improve classification accuracy. Inspired from this work, we have used *shrinkage* to handle partially specified text data where a multinomial model is assumed as the underlying generative model for the text documents.

6.1. Discussion

Due to the unavailability of data sources that are already annotated with relevant meta data, we performed experiments on only one data set, the relational Cora data set [9] . In our experiments, we manually associated an abstraction hierarchy over values of the attribute `Article`.*Words*. The hierarchy over the values of the attribute `Article`.*Topic* is provided by the Cora data set.

The algorithm presented here assumes a pre-specified *level of abstraction* defined by a global cut through the ontology. Our experiments have shown that the choice of the level of abstraction can impact the performance of the classifier. This suggests the possibility of improving the algorithm using a top down, iterative approach to refining the cut (see Figure 2b), starting with the most abstract cut in the abstraction hierarchy corresponding to the `Article`.*Words* attribute until an "optimal cut" and, thus, an optimal level of abstraction is identified for the learning task at hand. This strategy is similar to that adopted in [19] in learning Naïve Bayes classifiers from a single *flat* table, in the presence of attribute value taxonomies and partially specified data.

We investigated the effect of partially specified data on the performance of classifiers designed to predict the topic of a text document. Our experiments have shown that incorporating the effect of partially specified data through the means of cumulative term frequency counts (i.e., *shrinkage*) in computing the statistics used by the learning algorithm improves the performance of the resulting classifiers.

Some directions for future research include: implementation and experimental evaluation of a large class of algorithms for learning predictive models from structured relational data sources annotated with relevant meta data, including multi-modal data (e.g., images); learning classifiers from a collection of semantically disparate, structured relational data sources, each annotated with meta data; exploring the effect of using different ontologies and mappings in a distributed setting; the effect of degree of incompleteness of mappings; the effects of errors in mappings, etc.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[2] D. Caragea, J. Bao, and V. Honavar. Learning relational bayesian classifiers on the semantic web. In *Proceedings of the IJCAI 2007 SWeCKa Workshop*, India, 2007.

[3] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of relational structure. *Journal of Machine Learning Research*, 3:679–707, December 2002.

[4] W. James and C. Stein. Estimation with quadratic loss. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, pages 361–379, University of California Press, 1961.

[5] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. Wiley, 2002.

[6] Q. Lu and L. Getoor. Link-based classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2003.

[7] J. C. Mallery. Semantic content analysis: A new methodology for the relatus natural language environment. 1991.

[8] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification, 1998.

[9] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the contruction of internet portals with machine learning. *Information Retrieval Journal*, 3:127–163, 2000.

[10] A. K. McCallum, R. Rosenfeld, T. M. Mitchell, and A. Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In J. W. Shavlik, editor, *Proceedings of ICML-98*, pages 359–367, Madison, US, 1998.

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

[12] J. Neville, D. Jensen, and B. Gallagher. Simple estimators for relational bayesian classifiers. In *Proceedings of the Third IEEE International Conference on Data Mining*. IEEE Press, 2003.

[13] S. Rajan, K. Punera, and J. Ghosh. A maximum likelihood framework for integrating taxonomies. In *Proceedings of AAAI, Pittsburgh, Pennsylvania, USA*, pages 856–861, 2005.

[14] C. Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, pages 197–206, University of California Press, 1955.

[15] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proc. Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, Edmonton, Canada, 2002.

[16] B. Taskar, E. Segal, and D. Koller. Probabilistic supervised learning and clustering in relational data. In *Proceedings of IJCAI*, pages 870–876, Seattle, Washington, 2001.

[17] L. Valiant. A theory of the learnable. *Communications of the Association for Computing Machinery*, 27:1134–1142, 1984.

[18] J. Zhang and V. Honavar. Learning decision tree classifiers from attribute-value taxonomies and partially specified data. In T. Fawcett and N. Mishra, editors, *Proceedings of the International Conference on Machine Learning*, pages 880–887, Washington, DC, 2003.

[19] J. Zhang, D.-K. Kang, A. Silvescu, and V. Honavar. Learning compact and accurate naive bayes classifiers from attribute value taxonomies and data. *Knowledge and Information Systems*, 2005.