2005

Multinomial Event Model Based Abstraction for Sequence and Text Classification

Dae-Ki Kang
Iowa State University

Jun Zhange
Iowa State University

Adrian Silvescu
Iowa State University

Vasant Honavar
Iowa State University

Recommended Citation

Kang, Dae-Ki; Zhange, Jun; Silvescu, Adrian; and Honavar, Vasant, "Multinomial Event Model Based Abstraction for Sequence and Text Classification" (2005). *Computer Science Technical Reports*. 238.
http://lib.dr.iastate.edu/cs_techreports/238

Multinomial Event Model Based Abstraction for Sequence and Text Classification

Abstract

In many machine learning applications that deal with sequences, there is a need for learning algorithms that can effectively utilize the hierarchical grouping of words. We introduce Word Taxonomy guided Naive Bayes Learner for the Multinomial Event Model (WTNBL-MN) that exploits word taxonomy to generate compact classifiers, and Word Taxonomy Learner (WTL) for automated construction of word taxonomy from sequence data. WTNBL-MN is a generalization of the Naive Bayes learner for the Multinomial Event Model for learning classifiers from data using word taxonomy. WTL uses hierarchical agglomerative clustering to cluster words based on the distribution of class labels that co-occur with the word counts. Our experimental results on protein localization sequences and Reuters text show that the proposed algorithms can generate Naive Bayes classifiers that are more compact and similar or often more accurate than those produced by standard Naive Bayes learner for the Multinomial Model.

Disciplines

Artificial Intelligence and Robotics

Multinomial Event Model Based Abstraction for Sequence and Text Classification

Dae-Ki Kang, Jun Zhang, Adrian Silvescu, and Vasant Honavar

Artificial Intelligence Research Laboratory
Department of Computer Science
Iowa State University
Ames, IA 50011 USA
{dkkang, jzhang, silvescu, honavar}@cs.iastate.edu

Abstract. In many machine learning applications that deal with sequences, there is a need for learning algorithms that can effectively utilize the hierarchical grouping of words. We introduce Word Taxonomy guided Naive Bayes Learner for the Multinomial Event Model (WTNBL-MN) that exploits word taxonomy to generate compact classifiers, and Word Taxonomy Learner (WTL) for automated construction of word taxonomy from sequence data. WTNBL-MN is a generalization of the Naive Bayes learner for the Multinomial Event Model for learning classifiers from data using word taxonomy. WTL uses hierarchical agglomerative clustering to cluster words based on the distribution of class labels that co-occur with the word counts. Our experimental results on protein localization sequences and Reuters text show that the proposed algorithms can generate Naive Bayes classifiers that are more compact and similar or often more accurate than those produced by standard Naive Bayes learner for the Multinomial Model.

1 Introduction

In machine learning, one of the important goals is to induce comprehensible, yet accurate and robust classifiers [1]. In classical inductive learning for text classification, each document is represented as a bag of words. That is, one instance is an ordered tuple of word frequencies or binary values to denote the presence of words. However, these words can be grouped together to reflect assumed or actual similarities among the words in the domain or in the context of a specific application. Such a hierarchical grouping of words yields word taxonomy (WT). Figure 1 is an example of word taxonomy of “Science” made by human.

Word taxonomies are very common and useful in many applications. For example, Gene Ontology Consortium has developed hierarchical taxonomies for describing various aspects of macromolecular sequences, structures, and functions [2]. For intrusion detection, Undercoffer et al.[3] established a hierarchical taxonomy of features observable by the target of an attack. Various ontologies have been developed in fields related with Semantic Web [4].

Word taxonomies present the possibility of learning classification rules that are simpler and easier-to-understand when the terms in the rules are expressed

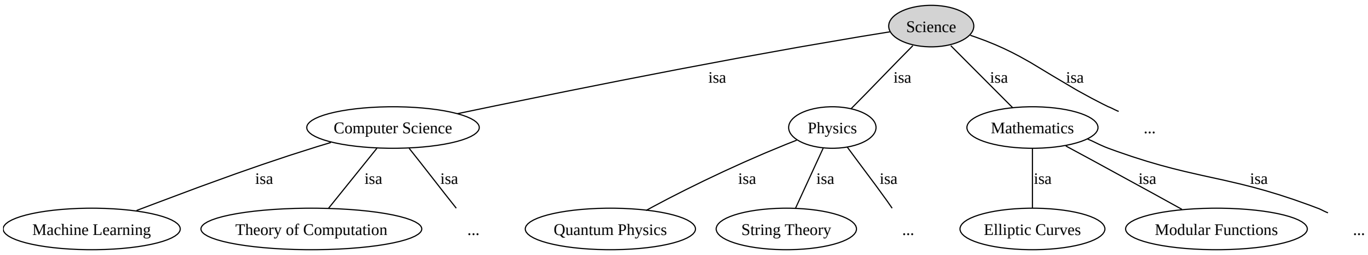

Fig. 1. Illustrative taxonomy of 'Science' by human

in terms of abstract values. Kohavi and Provost [5] pointed the need of incorporating hierarchically structured background knowledge. Abstraction of similar concepts by the means of attribute value taxonomy (AVT) has been shown to be useful in generating concise and accurate classifiers [6–8]. Zhang and Honavar [8] presented AVT-NBL, an algorithm that exploits AVTs to generate Naive Bayes Classifiers that are more compact and often more accurate than classifiers that do not use AVTs. The algorithm potentially performs regularization to minimize over-fitting from learning with relatively small data sets.

Against this background, we introduce word taxonomy guided Naive Bayes learner for the multinomial event model (WTNBL-MN). WTNBL-MN is a word taxonomy based generalization of the standard Naive Bayes learning algorithm for the multinomial model.

We also introduce word taxonomy learner (WTL) that automatically generates word taxonomy from sequence data by clustering of words based on their class conditional distribution. Because word taxonomy is not available in many domains, there is a need for automated construction of word taxonomy.

To evaluate our algorithms, we conducted experiments using two classification tasks: (a) assigning Reuters newswire articles to categories, (b) and classifying protein sequences in terms of their localization. We used Word Taxonomy Learner (WTL) to generate word taxonomy from the training data. The generated word taxonomy was provided to WTNBL-MN to generate concise Naive Bayes classifiers that used abstract words of word taxonomy.

The rest of this paper is organized as follows: Section 2 introduces the WTNBL-MN algorithm; Section 3 presents WTL algorithm; Section 4 describes our experimental results and Section 5 concludes with summary and discussion.

2 Word Taxonomy guided Naive Bayes Learner for the Multinomial Event Model (WTNBL-MN)

We start with definitions of preliminary concepts necessary to describe our algorithms. We then precisely define the problem as learning classifier from word taxonomy and sequence data.

2.1 Word Taxonomy

Let $\Sigma = \{w_1, w_2, \ldots, w_N\}$ be a set of words, $C = \{c_1, c_2, \ldots, c_M\}$ a finite set of mutually disjoint class labels, and $f_{i,j}$ denote an integer frequency of word w_i in a sequence d_j. Then, sequence d_j is represented as an instance I_j, a frequency vector $< f_{i,j} >$ of w_i, and each sequence belongs to a class label in C. Finally, a data set D is represented as a collection of instance and their associated class label (I_j, c_j).

Let T_Σ be an word taxonomy defined over the possible words of Σ. Let $Nodes(T_\Sigma)$ denote the set of all values in T_Σ and $Root(T_\Sigma)$ denote the root of T_Σ. We represent the set of leaves of T_Σ as $Leaves(T_\Sigma) \subseteq \Sigma$. The internal nodes of the tree correspond to *abstract values* of Σ.

After Haussler [9], we define a cut γ for word taxonomy T_Σ as follows.

Definition 1 (Cut) *A cut γ is a subset of nodes in word taxonomy T_Σ satisfying the following two properties:*

1. *For any leaf $l \in Leaves(T_\Sigma)$, either $l \in \gamma$ or l is a descendant of a node in T_Σ.*
2. *For any two nodes f,g $\in \gamma$, f is neither a descendant not an ancestor of g.*

A cut γ induces a partition of words in T_Σ. For example, in figure 1, a cut $\{ComputerScience, Physics, Mathematics\}$ defines a partition over the primitive words of 'Science' domain.

Definition 2 (Refinement) *We say that a cut $\hat{\gamma}$ is a refinement of a cut γ if $\hat{\gamma}$ is obtained by replacing at least one node $v \in \gamma$ by its descendants. Conversely, γ is an abstract of $\hat{\gamma}$*

Figure 2 illustrates a refinement process in word taxonomy T_Σ. The cut $\gamma = \{A, B\}$ is been refined to $\hat{\gamma} = \{A_1, A_2, B\}$ by replacing B with A_1 and A_2. Thus, corresponding hypothesis $h_{\hat{\gamma}}$ is a refinement of h_γ.

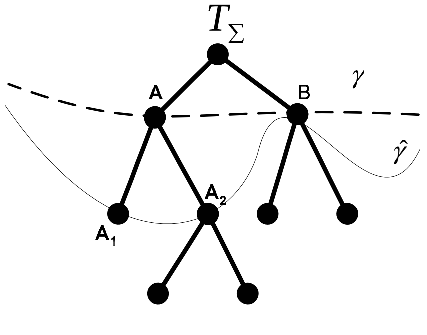

Fig. 2. Illustration of Cut Refinement: The cut $\gamma = \{A, B\}$ is been refined to $\hat{\gamma} = \{A_1, A_2, B\}$ by replacing B with A_1 and A_2

Definition 3 (Instance Space) *Any choice of γ defines an input space $\mathscr{I}_\gamma$. If there is a node $\in \gamma$ and $\notin$ Leaves(T_Σ), the induced input space $\mathscr{I}_\gamma$ is an abstraction of the original input space $\mathscr{I}$.*

With a data set D, word taxonomy T_Σ and corresponding valid cuts, we can extend our definition of instance space to include instance spaces induced from different levels of abstraction of the original input space. Thus, word taxonomy guided learning algorithm work on this induced input space.

2.2 Event Models for Naive Bayes Sequence Classification

WTNBL-MN algorithm generates Naive Bayes Classifier for the multinomial model. Before we describe WTNBL-MN algorithm, we briefly summarize event models for Naive Bayes classification. There are two popular event models [10, 11] for Naive Bayes classification of sequence data.

Multi-variate Bernoulli model In multi-variate Bernoulli model, a sequence d_j is represented as an instance I_j by a vector of binary values $b_{i,j} \in \{0,1\}$ to denote presence of a word w_i in the sequence. The number of occurrence of word is not preserved in the vector. The probability of sequence d_j given its class c_j is as follows:

$$P(d_j|c_j) = \prod_{i=1}^{|\Sigma|} (b_{i,j}p_{i,j} + (1 - b_{i,j})(1 - p_{i,j})) \tag{1}$$

Multinomial model In multinomial model, a sequence is considered as a vector of integer occurrences of word $f_{i,j}$. The probability of an instance I_j given its class c_j is defined as follows:

$$P(d_j|c_j) = \left\{ \frac{\left(\sum_i^{|\Sigma|} f_{i,j}\right)!}{\prod_i^{|\Sigma|}(f_{i,j})!} \right\} \prod_i^{|\Sigma|} \{p_{i,j}^{f_{i,j}}\} \tag{2}$$

The term $\left\{ \frac{\sum_i^{|\Sigma|} f_{i,j} \ !}{\prod_i^{|\Sigma|}(f_{i,j})!} \right\}$ represents the number of possible combinations of words for the instance I_j.

In equation 2, $p_{i,j}$ is basically calculated as follows:

$$p_{i,j} = \frac{Count(c_j, w_i)}{Count(c_j)}$$

$Count(c_j, w_i)$ is the number of times word w_i appears in all the instances that have a class label c_j, and $Count(c_j)$ is the total number of words in a particular class label c_j.

2.3 WTNBL-MN Algorithm

The problem of learning classifiers from word taxonomy and sequence data is a natural generalization of the problem of learning classifiers from the sequence data. Original data set D is a collection of labeled instances $< I_j, c_j >$ where $I \in \mathscr{I}$. A classifier is a hypothesis in the form of function $h : \mathscr{I} \rightarrow C$, whose domain is the instance space $\mathscr{I}$ and whose range is the set of class C. A hypothesis space $\mathscr{H}$ is a set of hypotheses that can be represented in some hypothesis language or by a parameterized family of functions. Then, the task of learning classifiers from the data set D is induce a hypothesis $h \in \mathscr{H}$ that satisfies given criteria.

Hence, the problem of learning classifiers from word taxonomy and data can be described as follows: Given word taxonomy T_Σ over words Σ and a data set D, the aim is induce a classifier $h_{\gamma^*} : \mathscr{I}_{\gamma^*} \rightarrow C$ where γ^* is a cut that maximizes given criteria. Of interest in this paper is that the resulting hypothesis space $\mathscr{H}_{\hat{\gamma}}$ of a chosen cut $\hat{\gamma}$ is efficient in searching for both concise and accurate hypothesis.

Word taxonomy guided Naive Bayes Learner is composed of two major components: (a) estimation of parameters of Naive Bayes classifiers based on a cut, (b) and a criterion for refining a cut.

Aggregation of Class Conditional Frequency Counts Parameter estimation can be efficiently done by aggregating class conditional frequency counts. For a particular node of a given cut, parameters of the node can be estimated by summing up the class conditional frequency counts of its children.

Given word taxonomy T_Σ, we can define a tree of class conditional frequency counts T_f such that there is one-to-one correspondence between the nodes of word taxonomy T_Σ and the nodes of the corresponding T_f. The class conditional frequency counts associated with a non leaf node of T_f is the aggregation of the corresponding class conditional frequency counts associated with its children. Because a cut through word taxonomy corresponds a partition of the set of words, the corresponding cut through T_f specifies a valid class conditional probability table for words. Therefore, to estimate each nodes of T_f, we simply estimate the class conditional frequency counts of primitive words in Σ, which corresponds to the leaves of T_f. Then we aggregate them recursively to calculate the class conditional frequency counts associated with their parent node.

Conditional Minimum Description Length of Naive Bayes Classifier For the criterion of hypothesis selection, we employed conditional minimum description length (CMDL) [12] for Naive Bayes classifier for the multinomial model.

Let v_j be a set of attribute values of j^{th} instance $d_j \in D$, and $c_j \in C$ a class label associated with d_j. Then, the conditional log likelihood of the hypothesis B given data D is

$$CLL(B|D) = |D| \sum^{|D|} \log\{P_B(c|v)\} = |D| \sum^{|D|} \log \left\{ \frac{P_B(c) P_B(v|c)}{\sum_{c_i}^{|C|} P_B(c_i) P_B(v|c_i)} \right\} \tag{3}$$

For Naive Bayes classifier, this score can be efficiently calculated [8].

$$CLL(B|D) = |D| \sum^{|D|} \log \left\{ \frac{P_B(c) \prod^{v_i \in v} \{P_B(v_i|c)\}}{\sum_{c_i}^{|C|} P_B(c_i) \prod^{v_j \in v} \{P_B(v_j|c_i)\}} \right\}$$

And the corresponding conditional minimum description length (CMDL) score is defined as follows:

$$CMDL(B|D) = -CLL(B|D) + \left\{ \frac{\log |D|}{2} \right\} size(B)$$

where, $size(B)$ is a size of the hypothesis B which corresponds to the number of entries in conditional probability tables (CPT) of B.

In case of a Naive Bayes classifier with multi-variate Bernoulli model, $size(B)$ is defined as

$$size(B) = |C| + |C| \sum_{i=1}^{|v|} |v_i|$$

where $|C|$ is the number of class labels, $|v|$ is the number of attributes, and $|v_i|$ is the number of attribute values for an attribute v_i.

Conditional Minimum Description Length of Naive Bayes Classifier for the Multinomial Model Combining the equations 2 and 3, we can obtain the conditional log likelihood of the classifier B given data D under the Naive Bayes multinomial model.

$$CLL(B|D) = |D| \sum_{j}^{|D|} \log \left\{ \frac{P(c_j) \left\{ \frac{\sum_i^{|\Sigma|} f_{i,j} \;!}{\prod_i^{|\Sigma|} (f_{i,j})!} \right\} \prod_i^{|\Sigma|} \{p_{i,j}^{f_{i,j}}\}}{\sum_k^{|C|} \left\{ P(c_k) \left\{ \frac{\sum_i^{|\Sigma|} f_{i,k} \;!}{\prod_i^{|\Sigma|} (f_{i,k})!} \right\} \prod_i^{|\Sigma|} \{p_{i,k}^{f_{i,k}}\} \right\}} \right\} \tag{4}$$

where, $|D|$ is the number of instances, $c_j \in C$ is a class label for instance $d_j \in D$, $f_{i,j}$ is a integer frequency of word $w_i \in \Sigma$ in instance d_j, and $p_{i,j}$ is the estimated probability that word w_i occurred in the instances associated to class label j.

Conditional Minimum Description Length (CMDL) of Naive Bayes Classifier for the multinomial model is defined as follows:

$$CMDL(B|D) = -CLL(B|D) + \left\{ \frac{\log |D|}{2} \right\} size(B)$$

where, $size(B)$ is a size of the hypothesis B which corresponds to the number of entries in conditional probability tables (CPT) of B.

Therefore, $size(B)$ is estimated as

$$size(B) = |C| + |C||v|$$

where $|C|$ is the number of class labels, $|v|$ is the number of attribute values.

Procedure Because each word is assumed to be independent of others given the class, the search for the word taxonomy guided Naive Bayes classifier can be performed efficiently by optimizing the CMDL criterion independently for each word. Thus, the resulting hypothesis h intuitively trades off the complexity in terms of the number of parameters against the accuracy of classification. The algorithm terminates when none of candidate refinements of the classifier yield statistically significant improvement in the CMDL score. Figure 3 outlines the algorithm.

```
WTNBL-MN:
begin

Input : data set $D$ and word taxonomy $T_\Sigma$
Initialize cut $\gamma$ to the root of $T_\Sigma$
Estimate probabilities that specify the hypothesis $h_\gamma$
Repeat until no change in cut $\gamma$
  $\bar{\gamma} \leftarrow \gamma$
  For each node $v \in \gamma$ :
    Generate a refinement $\gamma^v$ of $\gamma$ by replacing $v$ with its children.
    Construct corresponding hypothesis $h_{\gamma^v}$.
    If $CMDL(h_{\gamma^v}|D) < CMDL(h_{\bar{\gamma}}|D)$, then replace $\bar{\gamma}$ with $\gamma^v$.
  If $\gamma \neq \bar{\gamma}$ then $\gamma \leftarrow \bar{\gamma}$
Output : $h_\gamma$

end.
```

Fig. 3. Pseudo-code of Word Taxonomy Guided Naive Bayes Learner for the Multinomial Model(WTNBL-MN)

3 Learning Word Taxonomy from Sequence Data

We describe word taxonomy learner (WTL), a simple algorithm for automated construction of word taxonomy from sequence data.

3.1 Problem Definition

The problem of learning word taxonomy from sequence data can be stated as follows: Given a data set represented as a set of instances where an instance is a frequency vector $< f_i, c >$ of a word $w_i \in \Sigma$ and associated class label c, and a similarity measure among the words, output word taxonomy T_Σ such that it corresponds to a hierarchical grouping of words in Σ based on the specified similarity measure.

3.2 Algorithm

We use hierarchical agglomerative clustering (HAC) of words according to the distribution of class labels that co-occur with them. Let $DM(P(x)||Q(x))$ denote a measure of pairwise divergence between two probability distributions P and Q of the random variable x.

We use the pairwise divergence measure between the distribution of the class labels associated with the corresponding words as a measure of dissimilarity between the words. The lower the divergence between the class distribution between two words, the greater is their similarity. The choice of this measure of dissimilarity is motivated by the intended use of word taxonomy for WTNBL-MN algorithm to generate concise and accurate classifiers. If two words are indistinguishable from each other with respect to their class distribution, they will provide statistically similar information for classification of instance.

The algorithm of Word Taxonomy Learner (WTL) is shown in figure 4. The basic idea is to construct a taxonomy T_Σ by starting with the primitive words in Σ as the leaves of T_Σ and recursively add nodes to T_Σ one at a time by merging two existing nodes. To aid this process, the algorithm maintains a cut γ through the taxonomy T_Σ, updating the cut γ as new nodes are added to T_Σ. At each step, the two words to be grouped together to obtain an abstract word to be added to T_Σ are selected from γ based on the divergence between the class distributions associated with the corresponding words. That is, a pair of words in γ are merged if they have more similar class distributions than any other pair of words in γ. This process terminates when the cut γ contains a single word which corresponds to the root of T_Σ. The resulting T_Σ will have $(2|\Sigma| - 1)$ nodes when the algorithm terminates.

3.3 Similarity Measure

There are various ways to measure similarity between two probability distributions. We have tested thirteen different divergence measures. In our experiments, most of them resulted in similar performance on classification tasks.Hence, we limit the discussion to Jensen-Shannon divergence measure [13].

```
WTL:
begin

Input : data set D
For each word $w_i \in \Sigma$ :
  For each class $c_k \in C$ :
    Estimate the probability distribution $p\left(c_k|w_i\right)$
    Let $P\left(C|w_i\right) = \left\{p\left(c_1|w_i\right), \ldots, p\left(c_k|w_i\right)\right\}$ be the class distribution associated with the word $w_i$.
$\gamma \leftarrow \Sigma$;
Initialize $T_\Sigma$ with nodes in $\gamma$.
Iterate until $|\gamma| = 1$:
  In $\gamma$, find $(x, y) = argmin\left\{DM\left(P\left(C|x\right)||P\left(C|y\right)\right)\right\}$
  Merge $x$ and $y$ $(x \neq y)$ to create a new value $z$.
  Calculate probability distribution $P\left(C|z\right)$.
  $\hat{\gamma} \leftarrow \gamma \cup \{z\} \setminus \{x, y\}$.
  Update $T_\Sigma$ by adding nodes $z$ as a parent of $x$ and $y$.
  $\gamma \leftarrow \hat{\gamma}$.
Output : $T_\Sigma$

end.
```

Fig. 4. Pseudo-code of Word Taxonomy Learner (WTL)

Jensen Shannon Divergence is a weighted information gain that is reflexive, symmetric and bounded. Pairwise version of Jensen-Shannon divergence is given by

$$I\left(P||Q\right) = \frac{1}{2}\left[\sum p_i log\left(\frac{2p_i}{p_i + q_i}\right) + \sum q_i log\left(\frac{2q_i}{p_i + q_i}\right)\right]$$

4 Experiments

This section provides empirical evidences that WTNBL-MN coupled with WTL usually generate more concise and similar or often more accurate classifiers than those of the Naive Bayes classifiers for the multinomial model. The results are based on two different sequence data sets, text and protein. In each case, word taxonomy is generated using WTL and a classifier is constructed using WTNBL-MN on the resulting WT and sequence data.

4.1 Reuters 21587 Text Categorization Test Collection

Reuters 21587 distribution 1.0 data set consists of 12902 newswire articles in 135 overlapping topic categories. (This collection is publicly available at `http://www.daviddlewis.com/resources/testcollections/reuters21578/`.)

Following the previous works [14, 15, 10], we build binary classifiers for top ten most populous categories. In our experiment, stop words were not eliminated, and title words were not distinguished with body words. We conducted feature selection based on mutual information of features and selects top 300 features. The mutual information $MI(x, c)$ between a feature x and a category c is defined as:

$$MI(x, c) = \sum^{x} \left\{ \sum^{c} \left\{ P(x, c) log \frac{P(x, c)}{P(x)P(c)} \right\} \right\}$$

We followed the ModApte split in which 9603 stories are used for building classifiers and 3299 stories to test the accuracy of the resulting model. We report the break even points, the average of precision and recall when the difference between the two is minimum. Precision and recall of text categorization are defined as:

$$\text{Precision} = \frac{|\text{detected documents in the category (true positives)}|}{|\text{documents in the category (true positives + false negatives)}|}$$

$$\text{Recall} = \frac{|\text{detected documents in the category (true positives)}|}{|\text{detected documents (true positives + false positives)}|}$$

Table 1 shows the break even performance of precision and recall for the ten most frequent categories. WTNBL-MN usually shows similar performance in terms of break even performance except "corn" category, while the classifiers from WTNBL-MN has smaller size than those from Naive Bayes Learner (NBL).

Table 1. Break even Performance for 10 Largest Categories

Data	NBL-MN		WTNBL-MN		# of documents	
	breakeven	size	breakeven	size	train	test
earn	94.94	602	94.57	348	2877	1087
acq	89.43	602	89.43	472	1650	719
money-fx	64.80	602	65.36	346	538	179
grain	74.50	602	77.85	198	433	149
crude	79.89	602	76.72	182	389	189
trade	59.83	602	47.01	208	369	118
interest	61.07	602	59.54	366	347	131
ship	82.02	602	82.02	348	197	89
wheat	57.75	602	53.52	226	212	71
corn	57.14	602	21.43	106	182	56
Average (top 5)	80.71	602	80.79	309.2		
Average (top 10)	72.14	602	66.75	280		

Figure 5 shows Precision-Recall curve [16] of "grain" category. It can be seen that WTNBL-MN generates Naive Bayes classifier with smaller size that performs similarly to the classifier generated from Naive Bayes learner.

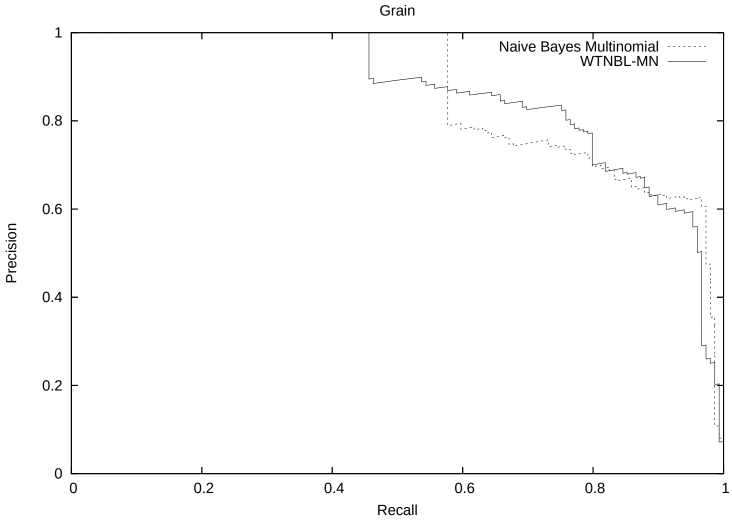

Fig. 5. Precision-Recall Curves of "Grain"

WTNBL-MN did not show good performance for "corn" category. It is because conditional log likelihood indicates accuracy of the model, not the optimal value of precision-recall for a particular class label. This results in stopping refinement of WTNBL-MN prematurely for the case one class label has less support, i.e. when the data set is imbalanced.

4.2 Protein Sequences

We applied WTNBL-MN algorithm on two protein data sets with a view to identifying their localization [17].

The first data set is 997 prokaryotic protein sequences derived from SWISS-PROT data base [18]. This data set includes proteins from three different subcellular locations: cytoplasmic (688 proteins), periplasmic (202 proteins), and extracellular (107 proteins). This dataset is available to download at `http://www.doe-mbi.ucla.edu/~astrid/astrid.html`.

The second data set is 2427 eukaryotic protein sequences derived from SWISS-PROT data base [18]. This data set includes proteins from the following four different subcellular locations: nuclear (1097 proteins), cytoplasmic (684 proteins), mitochondrial (321 proteins), extracellular (325 proteins). This dataset is available to download at `http://www.doe-mbi.ucla.edu/~astrid/astrid.html`.

For these data sets, we conducted ten-fold cross validation. To measure the performance of the following performance measures [19] are applied and the results for the data set are reported:

$$\text{Correlation coefficient} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP+FN})(\text{TP+FP})(\text{TN+FP})(\text{TN+FN})}}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP+TN+FP+FN}}$$

$$\text{Sensitivity}^{+} = \frac{\text{TP}}{\text{TP+FN}}$$

$$\text{Specificity}^{+} = \frac{\text{TP}}{\text{TP+FP}}$$

where, TP is the number of true positives, FP is the number of false positives, TN is the number of true negatives, and FN is the number of false negatives.

Figure 6 is a word taxonomy constructed for the prokaryotic protein sequences. Table 2 shows the results in terms of the performance measures for the two protein sequences. For both data sets, the classifier from WTNBL is more concise and shows similar or often more accurate performance in terms of the measures reported.

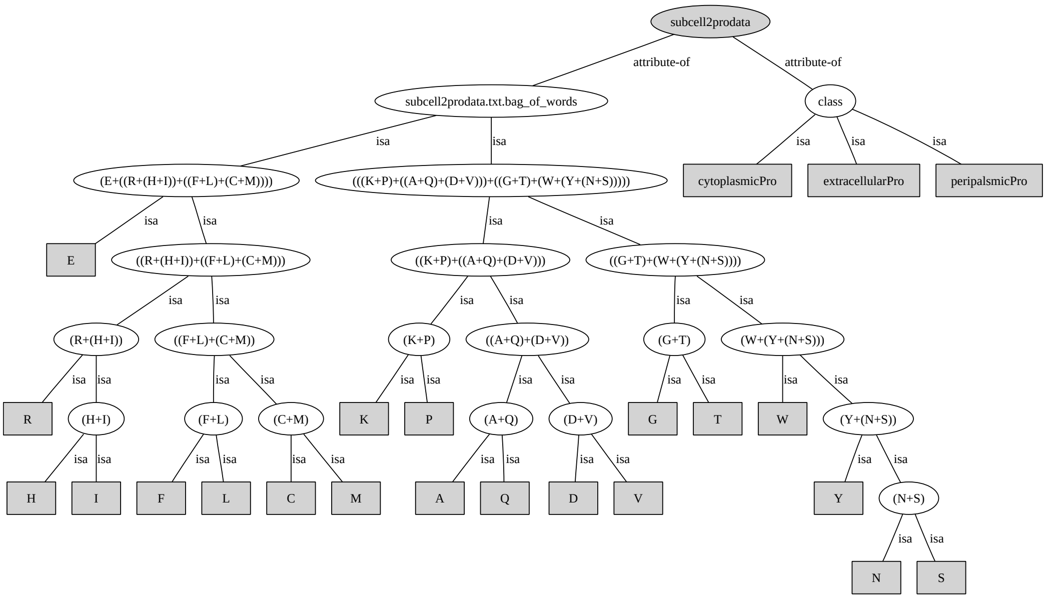

Fig. 6. Taxonomy from Prokaryotic Protein Localization Sequences constructed by WTL

Table 2. Results on Protein Localization Sequences (abbrev.: C - cytoplasmic, E - extracellular, P - peripalsmic, N - nuclear, M - mitochondrial)

Method	Prokaryotic			Eukaryotic			
	C	E	P	N	E	M	C
NBL-MN	71.96	70.57	51.31	61.00	36.83	25.13	44.05
correlation coefficient							
accuracy	88.26	93.58	81.85	80.72	83.11	71.69	71.41
specificity$^+$	89.60	65.93	53.85	82.06	40.23	25.85	49.55
sensitivity$^+$	93.90	83.18	72.77	73.38	53.85	61.06	81.29
size	42	42	42	46	46	46	46
WTNBL-MN	72.43	69.31	51.53	60.82	38.21	25.48	43.46
correlation coefficient							
accuracy	88.47	93.18	81.85	80.63	84.01	72.35	71.24
specificity$^+$	89.63	64.03	53.82	81.70	42.30	26.29	49.37
sensitivity$^+$	94.19	83.18	73.27	73.66	53.23	60.44	80.56
size	20	20	40	24	36	34	32

5 Summary and Related Work

5.1 Summary

Word taxonomy guided Naive Bayes Learning algorithm for the multinomial event model (WTNBL-MN) and automated word taxonomy learning algorithm (WTL) for sequence data are presented in this paper. WTNBL-MN is a generalization of the Naive Bayes learner for the multinomial event model for learning classifiers from data using word taxonomy. WTL is a hierarchical agglomerative clustering algorithm to cluster words into taxonomy based on the distribution of class labels that co-occur with the word counts. Experimental results on protein sequence and Reuters text show that the proposed algorithms can generate Naive Bayes classifiers that are more compact and similar or often more accurate than those produced by standard Naive Bayes learner for the Multinomial Model.

5.2 Related Work

There are some works in machine learning community on the problem of learning classifiers from attribute value taxonomies (AVT) or tree structured attributes.

Zhang and Honavar [6, 8] developed decision tree learner and Naive Bayes learner regularized over attribute value taxonomy. Their researches were primary focused on attribute value taxonomy for multi-variate data sets.

Taylor et al. [20] and Hendler et al. [21] described the use of taxonomy in rule learning. Han and Fu [22] proposed a method for exploiting hierarchically structured background knowledge for learning association rules. desJardins et al. [23]

suggested the use of Abstraction-Based-Search (ABS) to learning Bayesian networks with compact structure. To the best of our knowledge, there is no research about the regularization over word taxonomy for generating Naive Bayes classifiers for the multinomial model from a bag of words has not been investigated rigorously.

Gibson and Kleinberg [24] introduced STIRR, an iterative algorithm based on non-linear dynamic systems for clustering categorical attributes. Ganti et. al. [25] designed CACTUS, an algorithm that uses intra-attribute summaries to cluster attribute values. Both of them did not make taxonomies and use the generated for improving classification tasks.

Pereira et. al. [26] described distributional clustering for grouping words based on class distributions associated with the words in text classification. Slonim and Tishby [13] described a technique (called the agglomerative information bottleneck method) which extended the distributional clustering approach described by Pereira et al. [26], using Jensen-Shannon divergence for measuring distance between document class distributions associated with words and applied it to a text classification task. Baker and McCallum [27] reported improved performance on text classification using a distributional clustering with a Jensen-Shannon divergence measure. These works are mainly focused on clustering of words, but they did not apply the generated taxonomy for regularization to generate more concise classifiers.

5.3 Future Work

Some promising directions for future work include the following:

- Applying WTNBL-MN algorithm to up-to-date text corpora [28, 29].
- Enhancing WTNBL-MN and WTL algorithms for learning and exploiting hierarchical ontologies based on part-whole and other relations as opposed to ISA relations.
- Developing other measures for model selection rather than CMDL for cut refinement to accommodate the various application-specific needs.

References

1. Pazzani, M.J., Mani, S., Shankle, W.R.: Beyond concise and colorful: Learning intelligible rules. In: Knowledge Discovery and Data Mining. (1997) 235–238
2. Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., Harris, M., Hill, D., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J., Richardson, J., Ringwald, M., Rubin, G., Sherlock, G.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nature Genetics **25** (2000) 25–29
3. Undercoffer, J.L., Joshi, A., Finin, T., Pinkston, J.: A Target Centric Ontology for Intrusion Detection: Using DAML+OIL to Classify Intrusive Behaviors. Knowledge Engineering Review (2004)
4. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American (2001)

5. Kohavi, R., Provost, F.: Applications of data mining to electronic commerce. Data Mining and Knowledge Discovery **5** (2001) 5–10
6. Zhang, J., Honavar, V.: Learning decision tree classifiers from attribute value taxonomies and partially specified data. In: the Twentieth International Conference on Machine Learning (ICML 2003), Washington, DC (2003)
7. Kang, D.K., Silvescu, A., Zhang, J., Honavar, V.: Generation of attribute value taxonomies from data for data-driven construction of accurate and compact classifiers. In: Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), 1-4 November 2004, Brighton, UK. (2004) 130–137
8. Zhang, J., Honavar, V.: AVT-NBL: An algorithm for learning compact and accurate naive bayes classifiers from attribute value taxonomies and data. In: International Conference on Data Mining (ICDM 2004). (2004)
9. Haussler, D.: Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. Artificial intelligence **36** (1988) 177 – 221
10. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: AAAI-98 Workshop on Learning for Text Categorization. (1998)
11. Eyheramendy, S., Lewis, D.D., Madigan, D.: On the naive bayes model for text categorization. In: Ninth International Workshop on Artificial Intelligence and Statistics. (2003)
12. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Mach. Learn. **29** (1997) 131–163
13. Slonim, N., Tishby, N.: Agglomerative information bottleneck. In: Proceedings of the 13th Neural Information Processing Systems (NIPS 1999). (1999)
14. Dumais, S., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: CIKM '98: Proceedings of the seventh international conference on Information and knowledge management, ACM Press (1998) 148–155
15. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In Nédellec, C., Rouveirol, C., eds.: Proceedings of ECML-98, 10th European Conference on Machine Learning, Chemnitz, DE, Springer Verlag, Heidelberg, DE (1998) 137–142
16. Fawcett, T.: ROC graphs: Notes and practical considerations for researchers. Technical Report HPL-2003-4, HP Labs (2003)
17. Andorf, C., Silvescu, A., Dobbs, D., Honavar, V.: Learning classifiers for assigning protein sequences to gene ontology functional families. In: Fifth International Conference on Knowledge Based Computer Systems (KBCS 2004). (2004) 256–265
18. Bairoch, A., Apweiler, R.: The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Res. **28** (2000) 45–48
19. Yan, C., Dobbs, D., Honavar, V.: A two-stage classifier for identification of protein-protein interface residues. In: Proceedings Twelfth International Conference on Intelligent Systems for Molecular Biology / Third European Conference on Computational Biology (ISMB/ECCB 2004). (2004) 371–378
20. Taylor, M.G., Stoffel, K., Hendler, J.A.: Ontology-based induction of high level classification rules. In: DMKD. (1997) 0–
21. Hendler, J., Stoffel, K., Taylor, M.: Advances in high performance knowledge representation. Technical Report CS-TR-3672, University of Maryland Institute for Advanced Computer Studies Dept. of Computer Science (1996)
22. Han, J., Fu, Y.: Exploration of the power of attribute-oriented induction in data mining. In Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., eds.: Advances in Knowledge Discovery and Data Mining. AIII Press/MIT Press (1996)

23. desJardins, M., Getoor, L., Koller, D.: Using feature hierarchies in bayesian network learning. In: SARA '02: Proceedings of the 4th International Symposium on Abstraction, Reformulation, and Approximation, Springer-Verlag (2000) 260–270
24. Gibson, D., Kleinberg, J.M., Raghavan, P.: Clustering categorical data: An approach based on dynamical systems. VLDB Journal: Very Large Data Bases **8** (2000) 222–236
25. Ganti, V., Gehrke, J., Ramakrishnan, R.: Cactus - clustering categorical data using summaries. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press (1999) 73–83
26. Pereira, F., Tishby, N., Lee, L.: Distributional clustering of English words. In: 31st Annual Meeting of the ACL. (1993) 183–190
27. Baker, L.D., McCallum, A.K.: Distributional clustering of words for text classification. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press (1998) 96–103
28. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: A new benchmark collection for text categorization research. J. Mach. Learn. Res. **5** (2004) 361–397
29. Klimt, B., Yang, Y.: The Enron corpus: A new dataset for email classification research. In: 15th European Conference on Machine Learning (ECML2004). Vol. 3201 of Lecture Notes in Computer Science : Springer-Verlag. (2004) 217–226