RNBL-MN: A Recursive Naive Bayes Learner for Sequence Classification*

Dae-Ki Kang, Adrian Silvescu, and Vasant Honavar

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

Abstract. Naive Bayes (NB) classifier relies on the assumption that the instances in each class can be described by a *single* generative model. This assumption can be restrictive in many real world classification tasks. We describe RNBL-MN, which relaxes this assumption by constructing a tree of Naive Bayes classifiers for sequence classification, where each individual NB classifier in the tree is based on a multinomial event model (one for each class at each node in the tree). In our experiments on protein sequence and text classification tasks, we observe that RNBL-MN substantially outperforms NB classifier. Furthermore, our experiments show that RNBL-MN outperforms C4.5 decision tree learner (using tests on sequence composition statistics as the splitting criterion) and yields accuracies that are comparable to those of support vector machines (SVM) using similar information.

1 Introduction

Naive Bayes (NB) classifiers, due to their simplicity and modest computational and training data requirements, are among the most widely used classifiers on many classification tasks, including text classification tasks [1] and macromolecular sequence classification tasks that arise in bio-informatics applications [2]. NB classifiers belong to the family of generative models (a model for generating data given a class) for classification. Instances of a class are assumed to be generated by a random process which is modeled by a generative model. The parameters of the generative model are estimated (in the case of NB) assuming independence among the attributes given the class. New instances to be classified are assigned to the class that is the most probable for the instance.

NB classifier relies on the assumption that the instances in each class can be described by a *single* generative model (i.e., probability distribution). According to Langley [3], this assumption can be restrictive in many real world classification tasks. One way to overcome this limitation while maintaining some of the

* Supported in part by grants from the National Science Foundation (IIS 0219699) and the National Institutes of Health (GM 066387)

computational advantages of NB classifiers is to construct a tree of NB classifiers. Each node in the tree (a NB classifier) corresponds to one set of generative models (one generative model per class), with different nodes in the tree corresponding to different generative models for a given class. Langley described a recursive NB classifier (RBC) for classifying instances that are represented by ordered tuples of nominal attribute values. RBC works analogous to a decision tree learner [4], recursively partitioning the training set at each node in the tree until the NB classifier of the node simply cannot partition the corresponding data set. Unlike in the case of the standard decision tree, the branches out of each node correspond to the most likely class lebels assigned by the NB classifier at that node. In cases where each class cannot be accurately modeled by a single Naive Bayes generative model, the subset of instances routed to one or more branches belong to more than one class. RBC models the distribution of instances in a class at each node using a Naive Bayes generative model. However, according to Langley's reports of experiments on most of the UC-Irvine benchmark data sets, the recursive NB classifier did not yield significant improvements over standard NB classifier [3].

In this paper, we revisit the idea of recursive NB classifier in the context of sequence classification tasks. We describe RNBL-MN, an algorithm for constructing a tree of Naive Bayes classifiers for sequence classification. Each NB classifier in the tree is based on a multinomial event model [1] (one for each class at each node in the tree). Our choice of the multinomial event model is influenced by its reported advantages over the multivariate event model of sequences [1] in text classification tasks. RNBL-MN works in a manner similar to Langley's RBC, recursively partitioning the training set of labeled sequences at each node in the tree until a stopping criterion is satisfied. The branches out of each node correspond to the most likely class assigned by the NB classifier at that node. As for the stopping criterion, RNBL-MN uses a conditional minimum description length (CMDL) score for the classifier [5], specifically adapted to the case of RNBL-MN based on the CMDL score for the NB classifier using the multinomial event model for sequences [6]. Previous reports by Langley [3] in the case of a recursive NB classifier (RBC) for data sets whose the instances are represented as tuples of nominal attribute values (such as the UC-Irvine benchmark data), suggested that the tree of NB classifiers offered little improvement in accuracy over the standard NB classifier. In our experiments on protein sequence and text classification tasks, we observe that RNBL-MN substantially outperforms NB classifier. Furthermore, our experiments show that RNBL-MN outperforms C4.5 decision tree learner (using tests on sequence composition statistics as the splitting criterion) and yields accuracies that are comparable to those of SVM using similar information.

The rest of the paper is organized as follows: Section 2 briefly introduces the multinomial event model for sequences; Section 3 presents RNBL-MN (recursive Naive Bayes learner based on the multinomial event model for sequences); Section 4 presents our experimental results; Section 5 concludes with summary and discussion.

2 Multinomial Event Model for Naive Bayes Sequence Classification

Consider sequences defined over a finite alphabet $\Sigma = \{w_1 \cdots w_d\}$ where $d = |\Sigma|$. For example, in the case of protein sequences, Σ can be the 20-letter amino acid alphabet ($\Sigma = \{A_1, A_2, \ldots, A_{20}\}$). In the case of text, Σ corresponds to the finite vocabulary of words. Typically, a sequence $S_j \in \Sigma^\star$ is mapped into a finite dimensional feature space D through a mapping $\Phi : \Sigma^\star \rightarrow D$.

In a multinomial event model, a sequence S_j is represented by a *bag* of elements from Σ. That is, S_j is represented by a vector D_j of frequencies of occurrences in S_j of each element of Σ. Thus, $D_j =< f_{1j}, f_{2j}, \ldots, f_{dj}, c_j >$, where $f_{ij} \in \mathbb{Z}^*$ denotes the number of occurrences of w_i (the ith element of the alphabet Σ) in the sequence S_j. Thus, we can model the sequence S_j as a sequence of random draws from a multinomial distribution over the alphabet Σ. If we denote the probability of picking an element w_i given the class c_j by $P(w_i|c_j)$, the probability of sequence S_j given its class c_j under the multinomial event model is defined as follows:

$$P(X_1 = f_{1j}, \ldots, X_d = f_{dj}|c_j) = \left\{ \frac{\left(\sum_i^d f_{ij}\right)!}{\prod_i^d (f_{ij})!} \right\} \prod_{i=1}^{d} P(w_i|c_j)^{f_{ij}}$$

(Note: To be fully correct, we would need to multiply the right hand side of the above equation by $P(N|c_j)$, the probability of drawing a sequence of a specific length $N = (\sum_i^d f_{ij})$ given the class c_j, but this is hard to do in practice.)

Given a training set of sequences, it is straightforward to estimate the probabilities $P(w_i|c_j)$ using the Laplace estimator as $\hat{P}(w_i|c_j) = p_{ij} = \frac{Count_{ij}+1}{Count_j+d}$, where $Count_{ij}$ is the number of occurrences of w_i in sequences belonging to class c_j and $Count_j$ is the total number of words in training set sequences belonging to class c_j.

3 Recursive Naive Bayes Learner Based on the Multinomial Event Model for Sequences (RNBL-MN)

3.1 RNBL-MN Algorithm

As noted above, RNBL-MN, analogous to the decision tree learner, recursively partitions the training data set using Naive Bayes classifiers at each node of the tree. The root of the tree is a Naive Bayes classifier constructed from the entire data set. The outgoing branches correspond to the different class labels, assigned by the Naive Bayes classifier.

For a given input training data set $D_0 (= D_{current})$, we create a Naive Bayes classifier n_0. We compute the CMDL score $Score_{current}$ for the classifier n_0 (See section 3.2 for details of the calculation of CMDL score for recursive Naive Bayes classifier based on the multinomial event model). The classifier n_0 partitions the

data set D_0 into $|C|$ subsets based on the class labels assigned to the sequences by the classifier n_0. Each such subset is in turn used to train additional Naive Bayes classifiers. At each step, the CMDL score for the resulting tree of Naive Bayes classifiers is computed and compared with the CMDL score of the classifier from the previous step. This recursive process terminates when additional refinements of the classifier yield no significant improvement in CMDL score. Fig. 1 shows the pseudo-code of RNBL-MN algorithm.

RNBL-MN($D_{current}$) :
begin

1. **Input** : data set $D_0 = D_{current}$ // *data set*
2. Estimate probabilities given D_0 that specify the Naive Bayes classifier n_0
3. Add n_0 to the current classifier $h_{current}$ if $n_0 \notin h_{current}$
4. $Score_{current} \leftarrow CMDL(h_{current}|D_0)$ // *CMDL score of the current classifier*
5. Partition $D_{current}$ into $\mathbb{D} = \{D_1, D_2, \ldots, D_{|C|} | \forall_{S \in D_i} \forall_{j \neq i}, P(c_i|S) > P(c_j|S)\}$
6. For each $D_i \in \mathbb{D}$, estimate probabilities given D_i that specify the corresponding Naive Bayes classifiers n_i
7. $h_{potential} \leftarrow$ refinement of $h_{current}$ with the classifiers corresponding to each n_i based on the corresponding D_i in the previous step // *see Fig. 2 for details*
8. $Score_{potential} \leftarrow CMDL(h_{potential} | \sum_{i=0}^{|C|} D_i)$ // *CMDL score resulting from the refined classifier*
9. If $Score_{potential} > Score_{current}$ then // *accept the refinement*
10. Add each n_i to $h_{current}$
11. For each child node n_i
12. **RNBL-MN**(D_i) // *recursion*
13. End For
14. End If
15. **Output** : $h_{current}$

end.

Fig. 1. Recursive Naive Bayes Learner of Multinomial Event Model

Analogous to a decision tree, the resulting classifier predicts a class label for a new sequence as follows: starting at the root of the tree, the sequence is routed along the outgoing branches of successive Naive Bayes classifiers, at each node following the branch corresponding to the most likely class label for the sequence, until a leaf node is reached. The sequence is assigned the label corresponding to the leaf node.

3.2 Conditional Minimum Description Length (CMDL) score for Naive Bayes Classifier based on the Multinomial Event Model

RNBL-MN employs the conditional minimum description length (CMDL) score [5], specifically adapted to the case of RNBL-MN, based on the CMDL score for NB classifier using the multinomial event model for sequences [6] as the stopping criterion.

Recall the definition of a conditional minimum description length (CMDL) score of a classifier h given a data set D [5]:

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

where $size(h)$ is the size of the hypothesis h (the complexity of the model), which corresponds to the number of entries in the conditional probability tables (CPTs) of h. $CLL(h|D)$ is the conditional log likelihood of the hypothesis h given the data D, where each instance of the data has a class label $c \in C$.

When h is a Naive Bayes classifier based on a multinomial event model, the conditional log likelihood of the classifier h given data D can be estimated as follows [6]:

$$CLL(h|D) = |D| \sum_{j}^{|D|} \log \left\{ \frac{P(c_j) \left\{ \frac{(\sum_i^d f_{ij})!}{\prod_i^d (f_{ij})!} \right\} \prod_i^d \{p_{i,j}^{f_{ij}}\}}{\sum_k^{|C|} \left\{ P(c_k) \left\{ \frac{(\sum_i^d f_{ik})!}{\prod_i^d (f_{ik})!} \right\} \prod_i^d \{p_{i,k}^{f_{ik}}\} \right\}} \right\},$$

where $d = |\Sigma|$ is the cardinality of the vocabulary Σ, $|D|$ is the number of sequences in the data set D, $c_j \in C$ is the class label associated with the instance $S_j \in D$, f_{ij} is the integer frequency of element $w_i \in \Sigma$ in instance S_j, and $p_{i,j}$ is the estimated probability of the element w_i occurring in an instance belonging to class c_j.

The $size(h)$ for the multinomial event model is given by $size(h) = |C| + |C|d$, where $|C|$ is the number of class labels, and d is the cardinality of the vocabulary Σ.

3.3 CMDL for a Recursive Naive Bayes Classifier

We observe that in the case of a recursive Naive Bayes classifier, $CLL(h|D)$ can be decomposed in terms of the CLL scores of the individual Naive Bayes classifiers at the leaves of the tree of classifiers. Consequently, the CMDL score for the composite tree-structured classifier can be written as follows:

$$CMDL(h|D) = \sum_{node \in Leaves(h)} CLL(h_{node}|D_{node}) - \left\{\frac{\log|D|}{2}\right\} size(h),$$

where $size(h) = (|C| + |C|d)|h|$, $|h|$ denoting the number of nodes in h.

For example, Fig. 2 shows a Recursive Naive Bayes classifier consisting of 5 individual Naive Bayes classifiers. $\hat{c_+}$ and $\hat{c_-}$ are the predicted outputs of each hypothesis.

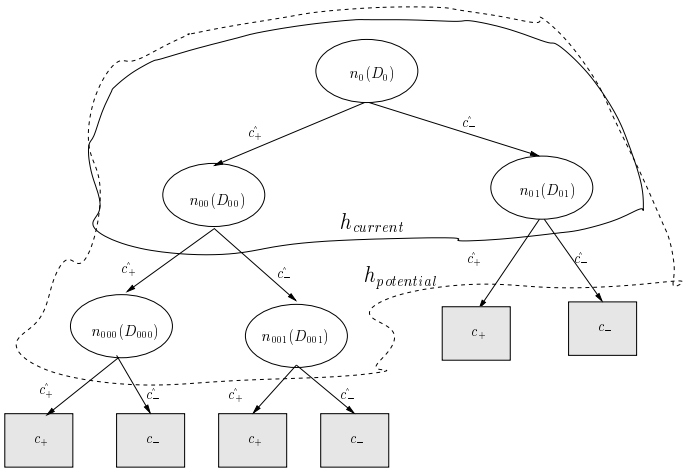

Fig. 2. Recursion tree of classifiers. Note that $h_{potential}$ is the refinement of $h_{current}$ by adding nodes $n_{000}(D_{000})$ and $n_{001}(D_{001})$ as children of $n_{00}(D_{00})$.

In the figure,

$$CLL(h_{current}|D) = CLL(n_{00}|D_{00}) + CLL(n_{01}|D_{01})$$

and

$$CLL(h_{potential}|D) = CLL(n_{000}|D_{000}) + CLL(n_{001}|D_{001}) + CLL(n_{01}|D_{01}),$$

where $|C|$=2 and $|h| = 5$.

Using the CMDL score, we can choose the hypothesis h that effectively trades off the complexity, measured by the number of parameters, against the accuracy of classification. As is described in Fig. 1, the algorithm terminates when none of the refinements of the classifier (splits of the tree nodes) yields statistically significant improvement in the overall CMDL score.

4 Experiments

To evaluate RNBL-MN, recursive Naive Bayes learner of multinomial event model, we conducted experiments using two classification tasks: (a) assigning Reuters newswire articles to categories, (b) and classifying protein sequences in terms of their cellular localization. The results of the experiments described in this section show that the classifiers generated by RNBL-MN are typically

more accurate than Naive Bayes classifiers using the multinomial model, and that RNBL-MN yields more accurate classifiers than C4.5 decision tree learner (using tests on sequence composition statistics as the splitting criterion). RNBL-MN yields accuracies that are comparable to those of linear kernel based SVM trained with the SMO algorithm [7] on a bag of letters (words) representation of sequences (text).

4.1 Reuters 21587 Text Categorization Test Collection

Reuters 21587 distribution 1.0 data set[1] consists of 12902 newswire articles in 135 overlapping topic categories. We followed the ModApte split [8] in which 9603 stories are used to train the classifier and 3299 stories to test the accuracy of the resulting classifier. We eliminated the stories that do not have any topic associated with them (i.e., no class label). As a result, 7775 stories were used for training and 3019 stories for testing the classifier.

Because each story has multiple topics (class labels), we built binary classifiers for the top ten most populous categories following the setup used in previous studies by other authors [9, 1]. In our experiments, stop words were not eliminated, and title words were not distinguished from body words. Following the widely used procedure for text classification tasks with large vocabularies, we selected top 300 features based on mutual information with class labels.

For evaluation of the classifiers, following the standard practice in text classification literature, we report the break-even points, which is the average of precision and recall when the difference between the two is minimum.

Table 1 shows the break-even points of precision and recall as a performance measure for the ten most frequent categories. The results in the table show that, RNBL-MN outperforms the other algorithms, except SVM, in terms of classification accuracy for Reuters 21587 text data set.

4.2 Protein Subcellular Localization Prediction

We applied RNBL-MN to two protein sequence data sets, where the goal is to predict the subcellular localization of the proteins [10, 2].

The first data set consists of 997 prokaryotic protein sequences derived from SWISS-PROT database (release 33.0) [11]. This data set includes proteins from three different subcellular locations: cytoplasmic (688 proteins), periplasmic (202 proteins), and extracellular (107 proteins).

The second data set contains 2427 eukaryotic protein sequences derived from SWISS-PROT database (release 33.0) [11]. This data set includes proteins from the following four different subcellular locations: nuclear (1097 proteins), cytoplasmic (684 proteins), mitochondrial (321 proteins), extracellular (325 proteins).

[1] This collection is publicly available at
http://www.daviddlewis.com/resources/testcollections/reuters21578/.

Table 1. Break-even point of precision and recall (a standard accuracy measure for ModApte split of Reuters 21587 data set) on the 10 largest categories of Reuters 21587 data set.

Data			NBL-MN	RNBL-MN	C4.5	SVM
name	# train (+/−)	# test (+/−)	accuracy	accuracy	accuracy	accuracy
earn	2877 / 4898	1087 / 1932	94.94	96.50	95.58	**97.24**
acq	1650 / 6125	719 / 2300	89.43	**93.32**	89.29	92.91
money-fx	538 / 7237	179 / 2840	64.80	69.83	69.27	**72.07**
grain	433 / 7342	149 / 2870	74.50	**89.26**	85.23	**89.26**
crude	389 / 7386	189 / 2830	79.89	77.78	76.19	**86.77**
trade	369 / 7406	117 / 2902	59.83	70.09	61.54	**71.79**
interest	347 / 7428	131 / 2888	61.07	70.99	64.89	**73.28**
ship	197 / 7578	89 / 2930	**82.02**	**82.02**	65.17	80.90
wheat	212 / 7563	71 / 2948	57.75	73.24	**87.32**	80.28
corn	181 / 7594	56 / 2963	57.14	67.85	**92.86**	76.79

The accuracy, sensitivity, and specificity of the classifiers (estimated using 10-fold cross-validation) on the two data sets [2] are shown in Table 2. The results show that RNBL-MN generally outperforms C4.5, and compares favorably with SVM. Specificity of SVM for 'Mitochondrial' is "N/A", because the SVM classifier always outputs negative when most of the instances in the data set have negative class label (imbalanced), which leads Specificity to be undefined.

5 Related Work and Conclusion

5.1 Related Work

As noted earlier, Langley [3] investigated recursive Bayesian classifiers for the instances described by tuples of nominal attribute values. RNBL-MN reported in this paper works with a multinomial event model for sequence classification.

Kohavi [12] introduced NBTree algorithm, a hybrid of a decision tree and Naive Bayes classifiers for instances represented using tuples of nominal attributes. NBTree evaluates the attributes available at each node to decide whether to continue building a decision tree or to terminate with a Naive Bayes classifier. In contrast, RNBL-MN algorithm, like Langley's RBC, builds a decision tree, whose nodes are all Naive Bayes Classifiers.

Gama and Brazdil [13] proposed an algorithm that generates a cascade of classifiers. Their algorithm combines Naive Bayes, C4.5 decision tree and linear discriminants, and introduces a new attribute at each stage of the cascade. They performed experiments on several UCI data sets [14] for classifying instances represented as tuples of nominal attribute values. In contrast, RNBL-MN recursively applies the Naive Bayes classifier based on the multinomial event model for sequences.

[2] These two datasets are available to download at http://www.doe-mbi.ucla.edu/~astrid/astrid.html.

Table 2. Localization prediction results on Prokaryotic and Eukaryotic protein sequences, calculated by 10-fold cross validation with 95% confidence interval.

(a) Prokaryotic protein sequences

Algorithm	Measure	Cytoplasmic	Extracellular	Peripalsmic
NBL-MN	accuracy	88.26±2.00	93.58±1.52	81.85±2.39
	specificity	89.60±1.89	65.93±2.94	53.85±3.09
	sensitivity	93.90±1.49	**83.18±2.32**	**72.77±2.76**
RNBL-MN	accuracy	**90.67±1.81**	**94.58±1.41**	**87.76±2.03**
	specificity	**91.61±1.72**	75.73±2.66	**73.53±2.74**
	sensitivity	95.20±1.33	72.90±2.76	61.88±3.01
C4.5	accuracy	84.15±2.27	91.98±1.69	84.65±2.24
	specificity	88.58±1.97	63.37±2.99	64.00±2.98
	sensitivity	88.32±1.99	59.81±3.04	55.45±3.09
SVM	accuracy	87.26±2.07	93.78±1.50	79.74±2.49
	specificity	84.67±2.24	**89.47±1.91**	50.00±3.10
	sensitivity	**99.56±0.41**	47.66±3.1	0.50±0.44

(b) Eukaryotic protein sequences

Algorithm	Measure	Cytoplasmic	Extracellular	Mitochondrial	Nuclear
NBL-MN	accuracy	71.41±1.80	83.11±1.49	71.69±1.79	80.72±1.57
	specificity	49.55±1.99	40.23±1.95	25.86±1.74	82.06±1.53
	sensitivity	**81.29±1.55**	53.85±1.98	**61.06±1.94**	73.38±1.76
RNBL-MN	accuracy	78.12±1.64	**92.13±1.07**	**87.72±1.31**	**83.48±1.48**
	specificity	60.24±1.95	75.97±1.70	**54.44±1.98**	84.30±1.45
	sensitivity	65.79±1.89	**60.31±1.95**	43.93±1.97	**78.09±1.65**
C4.5	accuracy	**78.99±1.62**	91.18±1.13	86.57±1.36	79.85±1.60
	specificity	63.51±1.92	69.89±1.83	49.03±1.99	77.94±1.65
	sensitivity	59.80±1.95	60.00±1.95	39.25±1.94	77.30±1.67
SVM	accuracy	71.98±1.79	86.69±1.35	86.77±1.35	79.36±1.61
	specificity	**83.33±1.48**	**100.00±0.00**	N/A	**87.53±1.31**
	sensitivity	0.73±0.34	0.62±0.31	0.00±0.00	63.35±1.92

5.2 Summary and Conclusion

RNBL-MN algorithm described in this paper relaxes the *single generative model per class* assumption of NB classifiers, while maintaining some of their computational advantages. RNBL-MN constructs a tree of Naive Bayes classifiers for sequence classification. It works in a manner similar to Langley's RBC [3], recursively partitioning the training set of labeled sequences at each node in the tree until a stopping criterion is satisfied. RNBL-MN employs the conditional minimum description length (CMDL) score for the classifier [5], specifically adapted to the case of RNBL-MN classifier based on the CMDL score for the Naive Bayes classifier using the multinomial event model [6] as the stopping criterion. Previous reports by Langley [3] in the case of a recursive NB classifier (RBC) on data sets whose instances were represented by tuples of nominal attribute values (such as the UC-Irvine benchmark data) had suggested that the tree of

NB classifiers offered little improvement in accuracy over the standard NB classifier. In contrast, we observe that on protein sequence and text classification tasks, RNBL-MN substantially outperforms the NB classifier. Furthermore, our experiments show that RNBL-MN outperforms C4.5 decision tree learner (using tests on sequence composition statistics as the splitting criterion) and yields accuracies that are comparable to those of SVM using similar information.

Given the relatively modest computational requirements of RNBL-MN relative to SVM, RNBL-MN is an attractive alternative to SVM in training classifiers on extremely large data sets of sequences or documents. Our results raise the possibility that Langley's RBC might outperform NB on more complex data sets in which the *one generative model per class* assumption is violated, especially if RBC is modified to use an appropriate CMDL criterion.

References

1. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: AAAI-98 Workshop on Learning for Text Categorization. (1998)
2. Andorf, C., Silvescu, A., Dobbs, D., Honavar, V.: Learning classifiers for assigning protein sequences to gene ontology functional families. In: 5^{th} International Conference on Knowledge Based Computer Systems. (2004) 256–265
3. Langley, P.: Induction of recursive bayesian classifiers. In: Proc. of the European Conf. on Machine Learning, London, UK, Springer-Verlag (1993) 153–164
4. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
5. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning **29** (1997) 131–163
6. Kang, D.K., Zhang, J., Silvescu, A., Honavar, V.: Multinomial event model based abstraction for sequence and text classification. In: 6^{th} International Symposium on Abstraction, Reformulation and Approximation. (2005) 134–148
7. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. Advances in kernel methods: support vector learning (1999) 185–208
8. Apté, C., Damerau, F., Weiss, S.M.: Towards language independent automated learning of text categorization models. In: 17^{th} annual international ACM SIGIR conference on Research and development in information retrieval. (1994) 23–30
9. Dumais, S., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: Proceedings of the 7^{th} international conference on Information and knowledge management, ACM Press (1998) 148–155
10. Reinhardt, A., Hubbard, T.: Using neural networks for prediction of the subcellular location of proteins. Nucleic Acids Research **26** (1998) 2230–2236
11. Bairoch, A., Apweiler, R.: The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Research **28** (2000) 45–48
12. Kohavi, R.: Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. In: Proc. of the 2^{nd} International Conference on Knowledge Discovery and Data Mining. (1996) 202–207
13. Gama, J., Brazdil, P.: Cascade generalization. Machine Learning **41** (2000) 315–343
14. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)