ISSN 1407-7493 COMPUTER SCIENCE DATORZINĀTNE 2008-7493

INFORMATION TECHNOLOGY AND MANAGEMENT SCIENCE
INFORMĀCIJAS TEHNOLOĢIJA UN VADĪBAS ZINĀTNE

A COMPARATIVE ANALYSIS OF CLASSIFICATION METHODS WITH INCREMENTAL LEARNING IN THE E-MAIL FILTERING TASK

Sigita Misina-Egle, Mg.sc.ing., Ph.D. student, Riga Technical University, Department of Modelling and Simulation, 1 Kalku Street, Riga, LV- 1658, Latvia, e-mail: sigita.misina@seb.lv.

Ludmila Aleksejeva, Dr.sc.ing., Assoc. Professor, Riga Technical University, Department of Modelling and Simulation, 1 Kalku Street, Riga, LV- 1658, Latvia, e-mail: ludmila.aleksejeva@cs.rtu.lv.

Keywords: incremental learning, e-mail classification, multilayer induction, dynamic window size

1. Introduction

For classifying objects in case of variable class description and on-line data it is appropriate to use incremental learning where learning examples are classified over time. Class description is not assumed to be constant in incremental learning and the algorithm learns incrementally by processing new data stream examples and forgetting old ones. This kind of learning has both advantages and disadvantages. The disadvantage is that in the course of time previously classified examples might not be classified correctly. But the advantage is that in case of knowledge aging incremental learning algorithm will pay more attention to the new class examples and will follow the context change.

This paper concerns research and analysis of several learning algorithms. The working principles of the algorithms are verified on a practical task of e-mail filtering.

2. E-mail classification task

The task of e-mail filtering is to create a classifier for new incoming e-mail classification into two classes: “forward” or “delete”– using the existing e-mail messages, which are not static, but on-line data.

The practical task is based on real e-mail correspondence from the forum about program *Lotus Notes* specific functions, restrictions and usage. All message texts are in Latvian, except for specific words – computer terms.

Electronic messages are described by four attributes [1] and class indication:

- *From* – sender e-mail address (without specific symbol @ and mail server);
- *Subject* – message subject, described with three representative words;
- *Body* – message body, described with four representative words;
- *Category* – message category, one of the following: question, answer, suggestion and information;
- *Classes* - agent activity: “forward” or “delete” new incoming message.

To create learning examples from e-mail messages, feature extraction algorithm has been used, where a feature means e-mail describing information: e-mail address, subject, and body. From field *From* all words have been taken. The average word count in field *Subject* is 6 words, but in field *Body* – 29 words. As the *Subject* usually contains e-mail message keywords, the 3 most common occurred words have been used. But field *Body* usually contains a lot of redundant information, so it is not necessary to review a large amount of words and in this case 4 most common occurred words have been taken. Every e-mail message has been processed so as to generate multiple learning set examples (see Figure 1).

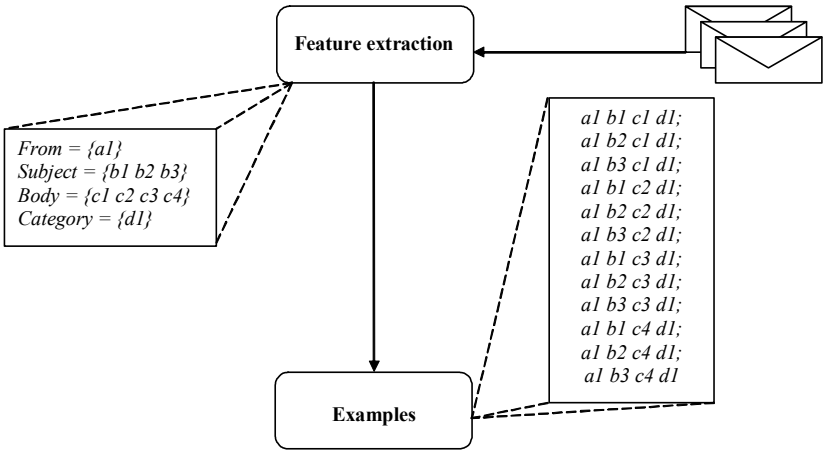

Figure 1. Learning example generation from feature set

In the example generation from e-mail messages, the *Levenshtein Distance* algorithm is used to get most common occurred words from *Subject* and *Body*. The *Levenshtein Distance* is a measure of the similarity between two strings, accordingly, the source string and the target string; the greater the *Levenshtein Distance*, the more different the strings are. A software *levenstain.jsp* that is based on *Levenshtein Distance* algorithm [2] is adapted and used for this particular task solution.

After processing 40 e-mail messages, 410 examples were acquired, where 160 belonged to class „delete" and 250 belonged to class „forward" [1], in proportion 39% to 61%.

In classification task when examples of a certain class predominate, to measure the rule accuracy, data of *confusion matrix* [3] are employed, whose size is $y*y$, where y is the count of classes. Table 1 shows a confidence matrix completed for the case of two classes when target class or class of interest is called *positive*, but the other class is called *negative*.

Table 1

Confusion matrix

		Predicted class		*Variables*
		+	-	
True class	+	$f_{++}(TP)$	$f_{+-}(FP)$	*TP* - True positive, correspond to the count of positive records;
				FP - False positive, correspond to the count of negative records but were classified as positive;
	-	$f_{-+}(FN)$	$f_{--}(TN)$	*FN*- False negative, correspond to the count of positive records but were classified as negative;
				TN -True negative, correspond to the count of negative records.

The confusion matrix reports classification outcomes, the count of correctly or incorrectly classified examples. To determine the accuracy, two measures are employed from the confusion matrix: positive class accuracy p, and recall r (see Expressions (1) and (2)):

$$p = TP/(TP + FP), \tag{1}$$

$$r = TP/(TP + FN), \tag{2}$$

In the practical experiments the positive class accuracy p is used to determine the classifier efficiency.

2.1. Inductive inference algorithm CN2 in mail interface agent

This paper describes mail agent interface - *MAGI* which stores user profile as a rule set to predict user behavior on incoming message distribution in folders. There are feature extraction and classification stages from *MAGI* architecture used in this paper. E-mail agent *MAGI* architecture model [5] clearly demonstrates agent working principles (see Figure 2).

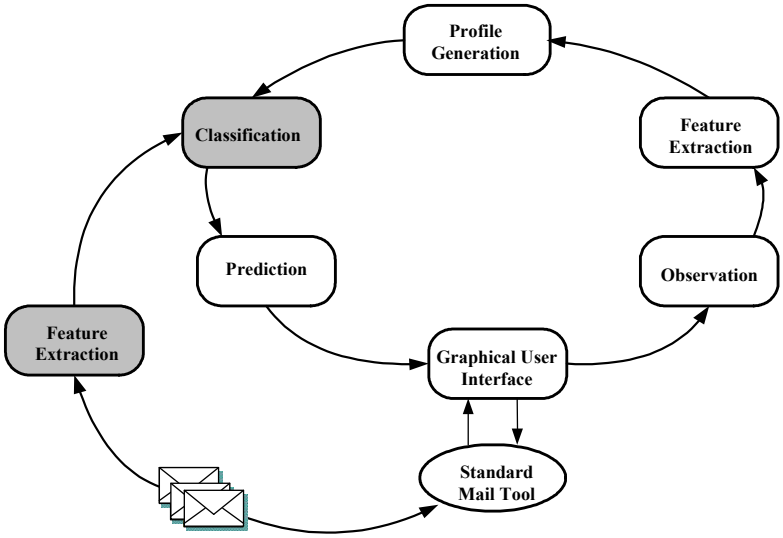

Figure 2. E-mail interface agent architecture

At first, an agent is taught using learning message set. Then the classification goes on, where classification in this context is activity which an agent has to perform on a message. The prediction stage evaluates the strength of classification (according to the method employed in the algorithm with which the classification is performed) for every new message and generates a confidence rating for it. Confidence rating for each activity is the number of rules obtained with one and the same action. Classification and confidence rating together form the agent's decision.

Mail agent interface *MAGI* is developed so that different methods and algorithms for rule construction, different methods for feature extraction and different prediction approaches can be used. There is inductive learning algorithm *CN2* [4] used in *MAGI* e-mail classification described in this paper. The *CN2* algorithm executes in an iterative fashion, searching in every iteration for such complex that covers a large count of examples from one class C_i and only few from other classes C_j where $C_i \neq C_j$. A complex is the conjunction constructed from values of attributes. When appropriate complex is found, the algorithm excludes the examples it covers from example set and adds rule „IF <complex> THEN <predict class *C*>" to the end

of rule list. The process continues until none of appropriate complexes can be found. Each new complex is evaluated and ranked using evaluation function, to determine its quality and significance [6]. The *Laplacian* error estimate function [7] can be used, which assesses the quality of the complex in making a classification:

$$LA = \frac{(n_c + 1)}{(n_{tot} + k)}, \tag{3}$$

where n_{tot} – total number of examples covered by the rule;
n_c – number of positive examples covered by the rule;
k – number of classes in the problem.

Practically using the *CN2* algorithm for e-mail message classification an experiment was made where the 5-fold cross-validation [8] method was used – all the data was randomly mixed and divided into 5 subsets of the same size (82 examples in each), where four subsets were used for learning (328 examples) and one for testing. Practical task was solved using freeware *CN2* version 6.1 [9]. Rules accuracy for common rule set is 83.60% in the experiment with cross-validation.

The initial motivation for using the *CN2* algorithm in agent learning process was that this algorithm generates human comprehensible rules by performing induction over training examples containing specific features. Yet, *CN2* is a static learning algorithm and despite driving it many times to imitate incremental learning in the e-mail message filtering task – which is a data stream task, more appropriate would be the incremental learning algorithm.

2.2. FLORA2 algorithm

As it was concluded that e-mail agent *MAGI* based on algorithm *CN2* is not effective for e-mail message filtering task, the incremental learning algorithm research was continued. As one of appropriate algorithms, the incremental learning algorithm *FLORA2* with adaptive example subset size adjustment heuristics has been found.

FLORA2 is one of the *FLORA* family algorithms [13]. *FLORA* algorithms do learning process by using three description sets. *ADES* is a set of all descriptions that are consistent (they match only positive examples). *PDES* is a set of candidate descriptions that, if taken together, are complete, but not consistent (it matches all positive examples, but also some negative ones) and *NDES* is a consistent description of the negative examples seen so far (it matches no positive examples).

Each description set can be interpreted as a disjunctive normal form (*DNF*) expression. *DNF* allows detecting if an expression is or is not inconsistent. *DNF* is disjunction of conjunctions [14]. Members of those conjunctions can be either true or false.

An example subset with which *FLORA2* works in the current step is called a window. In time, new examples are added to window while others are decided as old and forgotten [15]. The *FLORA2* algorithm [16] automatically detects and adjusts window size in the learning process. Basically, the idea is to reduce window size (and forget old examples) when it seems that the context drift can occur and keep the window size fixed in phases with a stable concept. Window size should increase while stable concept description is being made.

Dynamic window adjustment is performed according to a certain algorithm [17] where window size depends both on the count of covered positive examples *N* and on the size *S* of description set *ADES* (see Table 2). Parameter settings $lc = 1.2$ and $hc = 4$ and $p = 70\%$ are constants set by algorithm [17] authors.

Then two experiments have been performed where conjunctions by two and three attributes have been generated; all examples have been mixed 10 times to change their

sequence. In those practical experiments, incremental learning and forgetting using dynamic window size calculation have been performed.

The results of the previously made practical experiments [18] show that *FLORA2* generates a large number of rules in the form of conjunctions, also the "candidate" descriptions, which belong to both classes and could be useful for further learning process.

Table 2

Window size adjustment heuristics

Denotations:	
N ... number of positive instances covered by *ADES*	
S ... size of *ADES* in terms of number of literals	
Acc ... current predictive accuracy (monitored over recent classification attempts)	
|W| ... window size	
lc ... threshold for low coverage of *ADES*	
hc ... threshold for high coverage of *ADES*	
p ... threshold for acceptable predictive accuracy	
Algorithm:	
If (*N/S* < *lc*) vai ((*Acc* < *p*) un (decreasing *Acc*))	/* drift suspected */
then *L := 0.2 * |W|*	/* reduce window by 20% */
else if *(N/S > 2 * hc) un (Acc > p)*	/* extremely stable */
then *L := 2*	/* reduce window by 1 */
else if *(N/S > hc) un (Acc > p)*	/* stable enough */
then *L := 1*	/* keep window fixed */
then *L := 0*	/* increase window by 1 */

In new experiments, a large amount of rules was also gained - in the case of conjunctions by two, the average rule count gained was 182, but in the case of conjunctions by three – 292. As a result, the average rule classification accuracy was between 63% in case of conjunction by three and 89.5% in case of conjunction by two.

2.3. HMLII algorithm

As another appropriate incremental learning algorithm for e-mail message filtering task, the multilayer incremental induction algorithm *MLII* is researched and improved.

This paper describes a multilayer incremental inductive algorithm *MLII* hybrid *HMLII* [9] developed by the authors which is connected with the *CN2* [4] inductive inference algorithm. In the original version of *MLII* [10] the authors use the heuristic covering algorithm *HCV* [11] in the learning process, which is based on the extended matrix approach and as a result generates conjunctive formulas, and which is appropriate for practical task solving only with symbolic data. In its turn, the *CN2* algorithm generates human comprehensible rules and can process numeric as well as symbolic data (*HCV* works only with symbolic data). Due to that, to improve *MLII* performance, the hybrid algorithm *HMLII* [9] was developed in doctoral thesis where the *CN2* algorithm is used in the learning process.

The working principles of the multilayer incremental inference algorithm *MLII* [10] can be outlined in three steps:

A. Partition the initial data set into a number of layers (data subsets) of approximately equal size in a random shuffled way;
B. Learn a set of rules from the first subset of examples by generalization algorithm, get the rule set;

C. Perform the transition toward another learning problem, namely the refinement of the previous set of rules. This transition is performed by a redescription operator called reduction, which derives a new set of examples by examining the behavior of the rule set from the second stage over a second data subset.

Data partitioning (Stage A) dilutes noise in the original learning set and evenly distributes examples of different classes.

Generalization (Stage B) is used for initial information compressing. Generalization involves observing a subset of training examples of some particular concept, identifying examples, and then formulating a concept definition based on these common features. In *MLII*, discriminant generalization by elimination [10] is adapted. Generalization rule is a transformation of a description into a more general description.

Reduction (Stage C) derives a new set of behavioral examples by examining the behavior of the rule set from the previous step over a second data subset.

To investigate the multilayer incremental induction algorithm *MLII* and its hybrid *HMLII* working principles and efficiency in e-mail filtering task, three experiments have been done with the different layer count from three to five in each of them, researching additionally if the number of layers affects classification accuracy. Every experiment was organized into three steps:

1) Initial data processing to acquire basic learning data;
2) Three experiment performing based on learning data to generate classification rules;
3) Each experiment classifier testing.

From the first data subset or layer in learning process rules are generated, then the generalization and reduction operations are applied on them. In similar way other subsets are processed in all experiments. All rules gained from the subsets of experiment have been combined together (meta-rules were constructed) and the testing has been performed on the test set examples using *MS Excel* software.

The learning process has been done by free software *Sipina for Windows - Research version* [12] in the experiments.

The accuracy of the classifiers obtained as a result of *HMLII* hybrid algorithm learning, is considerably higher (see Table 3) as compared to solving the same e-mail filtering task using the original *MLII* algorithm.

Table 3

Summary of *MLII* and *HMLII* experiments

Layer count	*Accuracy (%)*	
	MLII	*HMLII*
3	80.65	79.31
4	62.71	92.59
5	75.44	84.62

Based on the experimental results, it can be concluded that the increase in the number of *HMLII* layers does not improve rule accuracy. The reduction working principles are left for further experiments, as the rules generated with algorithm *CN2* were simply with one attribute in condition part.

3. A comparative analysis of methods

Making a comparative analysis of the practical results obtained by the methods used, it can be concluded that the most appropriate for the e-mail filtering task is the *HMLII*

algorithm. The lowest accuracy can be observed (see Table 4) in the case of interface agent based on *CN2*. It can be due to the *CN2* algorithm is a static learning algorithm and was run many times to imitate incremental learning. The best results were shown by hybrid algorithm *HMLII*, where the testing accuracy of classifier gained in the learning process was between 79.31% and 92.59%. The *HMLII* algorithm uses multilayer incremental learning method with data partitioning, generalization and refinement. Using the original algorithm *MLII* accuracy was between 62.71% and 80.65%. The *FLORA2* algorithm generated the largest rule count in e-mail message filtering task, including also "candidate" rules that describe examples from both classes and could be useful for further classification process. However, classification accuracy of the *FLORA2* algorithm was only satisfactory – between 63% and 89.5%.

Table 4

Summary of experiments

Algorithm	Average rule count	Accuracy
CN2	73	83.60%
MLII	5	80.65%
FLORA2	127	89.50%
HMLII	10	92.59%

Comparing practical results (see Table 4) achieved by all three methods, the following can be concluded:

a) the multilayer incremental induction algorithm *HMLII* based on *CN2* has made a classifier that gives the best testing process result;
b) the *FLORA2* algorithm generates a large amount of rules, including „candidate" rules that belong to both classes and is useful for further classification process;
c) interface agent approach based on the inductive inference algorithm *CN2* requires plenty of expert time and is not effective.

4. Conclusions

A number of inductive classification algorithms with incremental learning are described, compared and practically realized in this paper. The e-mail message filtering task is performed using interface agent method; multilayer incremental induction algorithm *MLII*; multilayer incremental induction algorithm hybrid *HMLII* and incremental learning algorithm *FLORA2* with adaptive window size adjustment heuristic.

A comparative analysis of the results achieved in all practical experiments is made and conclusions about the algorithm efficiency and suitability for a particular task are drawn.

It is concluded that the *HMLII* algorithm best fits e-mail message classification; in the practical experiments it has shown the highest accuracy and generated not very large count of rules as compared to *FLORA2*, where a large count of rules was obtained as a result of learning.

Acknowledgments

We appreciate useful comments and suggestions on the research work made by Professor Arkady Borisov from Riga Technical University.

References

1. Misina S., Aleksejeva L. Inductive inference algorithms in e-mail messages filtering // 11th International Conference on Soft Computing MENDEL 2005. Brno, Czech Republic, 15-17 June 2005. – Brno: Brno University of Technology, 2005. P. 63 - 68. – ISBN 80-214-2961-5.
2. Gilleland M. Levenshtein Distance, in Three Flavors. Merriam Park Software. URL: http://www.merriampark.com/ld.htm. – Visit date November 2005.
3. Datu ieguve: Pamati / A. Sukovs, L. Aleksejeva, K. Makejeva, A. Borisovs. - Rīga: RTU, SIA "Drukātava", 2007. – 130 lpp.
4. Clark P., Niblett T. The CN2 Induction algorithm // Machine Learning. – 1989. -Vol. 3, No. 4. - P. 261 - 283.
5. Payne R. T., Edwards P. Interface Agents that Learn: An Investigation of Learning Issues in a Mail Agent Interface // Applied Artificial Intelligence, Vol. 11, No. 1, January 1997. - P. 1 - 32.
6. Han J., Kamber M. Data Mining: Concepts and Techniques. 2st Edition. – San Francisco etc.: Morgan Kaufman, 2006. – 800 P.
7. Clark P., Boswell R. Rule Induction with CN2: Some Recent Improvements // Machine Learning – EWSL-91: Proceedings of the 5th European Conference. - Berlin: Springer – Verlag, 1991. – P. 151 - 163.
8. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection // Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95). – San Mateo, CA: Morgan Kaufmann. – 1995. P. 1137 - 1143.
9. Clark P. Software: CN2 - Rule induction from examples. URL: http://www.cs.utexas.edu/users/pclark/software#cn2. – Visit date September 2004.
10. Wu X., Lo W.H.W. Multi-Layer Incremental Induction // Proceedings of the 5th Pacific Rim International Conference on Artificial Intelligence, Springer – Verlag, London, UK, 1998. - P. 24 - 32.
11. Wu X. Rule Induction with Extension Matrices // Journal of the American Society for Information Science. – Vol. 49, issue 5, 1998. - P. 435 - 454.
12. Sipina for Windows - Research version Laboratoire ERIC. URL: http://eric.univ-lyon2.fr/~ricco/sipina.html. – Visit date November 2005.
13. Misina S., Alexeyeva L. Efficiency analysis of on-line classification rule construction methods // Scientific Proceedings of Riga Technical University. Series 5. Computer science. Information technology and management science. Vol. 14 (2003) – P. 122 - 137. - ISSN 1407-7493.
14. Logic. An Introduction. Second Edition / Churchill R.P. - New York: St.Martin's Press, Inc., 1990. – 635 P.
15. Widmer G., Kubat M. Learning Flexible Concepts from Streams of Examples: FLORA2. // European Conference on Artificial Intelligence, ECAI-92, Vienna, Austria. 1992.
16. Widmer G. Combining robustness and flexibility in learning drifting concepts // Department of Medical Cybernetics and Artificial Intelligence, University of Vienna, and Austrian Research Institute for Artificial Intelligence. Issue 1, Machine Learning. – Vol.23, 1996. - P. 69 - 75.
17. Widmer G., Kubat M. Learning in the Presence of Concept Drift and Hidden Contexts // Machine Learning. - 1996. – Vol. 23. Issue 1, Kluwer Academic Publishers Hingham, MA, USA - P. 69 - 101. URL: http://www.miami.edu/eng-electrical/mkubat/Publications/germljfinal.ps. – Visit date September 2006.

18. Misina S. Example subset size adaptation heuristic in incremental learning // Scientific Proceedings of Riga Technical University. Series 5. Computer science. Information technology and management science. Vol. 28 (2006) – P. 107 - 114. - ISSN 1407-7493.

Misiņa-Egle Sigita, Aleksejeva Ludmila. Klasifikācijas metožu ar inkrementālu apmācību salīdzinošā analīze e-pasta ziņojumu filtrēšanas uzdevumā

Rakstā aprakstīti, salīdzināti un analizēti induktīvās klasifikācijas algoritmi ar inkrementālu apmācību e-pasta ziņojumu klasifikācijai. Šādi algoritmi ir efektīvi sfērās, kur tiek novērota zināšanu novecošanās, trokšņaini dati un klases apraksta nobīde. Kā arī šādi algoritmi ir piemēroti datu plūsmas apstrādei. Tika salīdzinātas šādas metodes: 1) interfeisa aģents MAGI lietots kopā ar induktīvās secināšanas algoritmu CN2; 2) daudzkārtainās inkrementālās secināšanas algoritms MLII; 3) pētījumu gaitā piedāvāts daudzkārtainās inkrementālās secināšanas algoritma hibrīds HMLII; 4) inkrementālās apmācības algoritms FLORA2 ar adaptīvu datu loga izmēra heiristiku. Praktiskajā e-pasta ziņojumu klasifikācijas uzdevumā viszemāko klasifikācijas precizitāti uzrādīja interfeisa aģenta metode – iemesls tam varētu būt statiskā algoritma CN2 izmantošana. Vislabākos rezultātus deva hibrīdais algoritms HMLII – precizitāte no 79,31% līdz 92,59%. Algoritms FLORA2 ģenerēja vislielāko likumu skaitu, iekļaujot arī kandidātu likumus, kas klasificē abu klašu piemērus, klasifikācijas precizitāte FLORA2 gadījumā iegūta no 63% līdz 89,5%. Tika secināts, ka piemērotākais e-pasta ziņojumu klasifikācijai ir algoritms HMLII, kurš praktiskajos eksperimentos uzrādīja vislabāko precizitāti un ģenerēja ne pārāk lielu skaitu likumu (salīdzinot ar FLORA2, kur apmācības rezultātā tika iegūts liels skaits likumu).

Misiņa-Egle Sigita, Aleksejeva Ludmila. A comparative analysis of classification methods with incremental learning in the e-mail filtering task

This paper describes compares and analyses inductive classification algorithms with incremental learning for e-mail classification. Such algorithms are effective in areas where knowledge aging is observed, where noisy data and concept drift are present. Also those algorithms are appropriate for data stream processing. The following methods have been compared: 1) interface agent MAGI used together with inductive inference algorithm CN2; 2) incremental learning algorithm FLORA2, which uses adaptive observation window size adaptation heuristics; 3) multilayer incremental inference algorithm MLII; 4) multilayer incremental inference algorithm hybrid HMLII. In practical e-mail classification task interface agent gives the lowest accuracy– the reason for that could be the usage of static algorithm CN2. The best results were provided by a hybrid algorithm HMLII – the accuracy was from 79.31% to 92.59%. Algorithm FLORA2 generated the largest count of rules, including candidate rules, which classify examples of both classes; the accuracy was good – from 63% to 89.5%. It is concluded that the HMLII algorithm best fits e-mail message classification; in the practical experiments it has shown the highest accuracy and generated not so many rules as compared to FLORA2, where a large count of rules was obtained as a result of learning.

Мисиня-Эгле Сигита, Алексеева Людмила. Сравнительный анализ методов классификации с инкрементальным обучением в задаче фильтрации почтовых сообщений

В статье описываются, сравниваются и анализируются индуктивные алгоритмы с инкрементальным обучением применительно к проблеме фильтрации почтовых сообщений. Подобные алгоритмы эффективны в сферах, где наблюдается устаревание информации, зашумленность данных и дрейф описания класса. Они также пригодны для обработки потоков данных. В работе сравниваются следующие методы: 1) агент взаимодействия MAGI совместно с алгоритмом индуктивного вывода CN2; 2) алгоритм многослойного инкрементального вывода MLII; 3) предложенный в ходе исследований гибрид алгоритма многослойного инкрементального вывода HMLII; 4) алгоритм инкрементального обучения FLORA2, с использованием эвристики адаптивного изменения размера окна данных. В практической задаче классификации электронных сообщений наиболее низкую точность показал агент взаимодействия – причиной этого может служить использование статического алгоритмя CN2. Наилучшие результаты у гибридного алгоритма HMLII – точность от 79,31% до 92,59%. Алгоритм FLORA2 генерирует большое количество правил, включая правила-кандидаты, которые классифицируют примеры обоих классов. Точность классификации лежит в пределах 63% - 89,5%. Таким образом, алгоритм HMLII является наиболее пригодным в задаче фильтрации почтовых сообщений, поскольку он показал наивысшую точность при небольшом числе сгенерированных правил (по сравнению с алгоритмом FLORA2).