World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:5, No:8, 2011

DWM-CDD: Dynamic Weighted Majority Concept Drift Detection for Spam Mail Filtering

Leili Nosrati and Alireza Nemaney Pour

***Abstract*—**Although e-mail is the most efficient and popular communication method, unwanted and mass unsolicited e-mails, also called spam mail, endanger the existence of the mail system. This paper proposes a new algorithm called Dynamic Weighted Majority Concept Drift Detection (DWM-CDD) for content-based filtering. The design purposes of DWM-CDD are first to accurate the performance of the previously proposed algorithms, and second to speed up the time to construct the model. The results show that DWM-CDD can detect both sudden and gradual changes quickly and accurately. Moreover, the time needed for model construction is less than previously proposed algorithms.

***Keywords*—**Concept drift, Content-based filtering, E-mail, Spam mail.

I. Introduction

TODAY, e-mail is the most commonly used forms of communication. Although, it is popular and efficient, unwanted and mass unsolicited e-mails, also called spam mail, endanger the existence of the mail system. According to reports, spam e-mail traffic has increased from 16% in 2002 to 80% in 2007 in North America. In addition, spam e-mails represent four out of every five e-mail messages today [1]. Consequently, people use a lot of time to get rid of them.

There are several major problems with spam mails. First of all, they are high in volume and fill in mailbox of users. Secondly, there is no correlation between receivers' area of interests and the contents of spam mails. Thirdly, they cost money for ISPs because the bandwidth and the memory of system are wasted. Finally, Spam e-mails cause a lot of security problems because most of them include Trojan, Malwares, and viruses [2].

There are many available techniques to detect and prevent the flow of spam e-mails. These techniques are categorized under the name of filtering. Generally, spam e-mail filtering is classified into two categorizes, rule-based [3] and content-based [4]. Rule-based filtering is similar to content based filtering with some differences. This technique works through some certain rules and regulations. By these rules the filter decides to pass or to block the received e-mail. Content-based filtering uses machine learning technique. In order to have the best results, the administrator of the mail server needs to train the filters to perform their functions.

These filtering techniques have restrictions. The problem with the rule-based filtering is that the rules and the policies need to be updated by the administrator of the system all the times. This work appears not to be an efficient and accurate work. The problem with content-base filtering is that the spammers are aware of filter techniques and the functionality, and may use additional characters to legitimize their e-mails. As rule-based filtering is not efficient and accurate, in addition the system needs to be updated frequently by the administrator; we choose content-based filtering for this work.

This paper proposes a new algorithm called Dynamic Weighted Majority Concept Drift Detection (DWM-CDD) for content-based filtering. The purposes of design of DWM-CDD are first to accurate the performance of the previously proposed algorithms, and second to speed up the time to construct the model. The results show that DWM-CDD can detect both sudden and gradual changes quickly and accurately. Moreover, the time needed for model construction is less than previously proposed algorithms.

The rest of this paper is organized as follows: section 2 discusses the researches based on concept drift. The design principles and detailed design of our proposal are shown in Sections 3 and 4 respectively. In Section 5, we show the experiment results, and compare our proposed algorithm with previously proposed ones. Finally, section 6 concludes the paper.

II. Related Work

In this section, we review the algorithms related to concept drift. Concept drift is defined as a part of an online learning task for changes in the concept of e-mails as time goes by. In other words, concept drift monitors the changes and the related implications in order to learn those changes. Authors in [2] classify the concept drift algorithms in details. STAGGER was the first system which addressed concept drift. This system uses a distributed concept description comprised of class nodes interrelated to attribute-value nodes through probabilistic arcs. Later, many algorithms like FLORA family and AQ-PM family followed STAGGER algorithm.

Concept versioning [2] is another concept drift system designed to cope with continuing evolutionary concept drifting. This system takes benefit from a frame representation, and manages such drifts by two methods; by altering current concept descriptions, or by making and creating a more recent version of these descriptions. The FLORA systems track concept drift by maintaining a sequence of examples over a dynamically adjusted window of time. It uses such examples to induce and refine three sets of rules; the rules covering the positive examples, the rules covering the negative examples, and the potential rules that are too general at present [5].

Leili Nousrati is with Dept. of the IT Engineering, Sharif University of Technology, International Campus, Kish Island, IRAN (e-mail: nosrati.leili@gmail.com).

Alireza Nemaney Pour is with Dept. of the IT engineering, Sharif University of Technology, International Campus, Kish Island, IRAN (e-mail: pour@sharif.edu).

Authors in [2] claim that meta-learning mechanisms can recognize contextual features. This can be achieved by analyzing the frequency and occurrence of a learner's entire history as well as a fixed window of time. FLORA [6] and AQ-PM [7] family with differences have this capability respectively. Starting from FLORA2, the algorithm can store the most recently encountered examples over a dynamically sized period of time. FLORA3 has mechanisms for coping with noise. Flora4 has extensions to deal with recurring contexts. On the other hand, AQ-PM uses the AQ algorithm to learn new rules from those ones stored in memory, and from new ones in the input stream. This algorithm forgets those rules after a fixed period of time. Although, AQ11-PM stores boundary examples, and like AQ-PM forgets them after a fixed period of time, it uses the AQ11 algorithm to form concepts incrementally. AQ-11-PM-Wah extends the algorithm of AQ11-PM. All of these systems have been evaluated on Stagger concepts.

According to [2], online algorithm is for training support vector machines. The special feature of this algorithm is that it adds the formerly obtained vectors to the recent training set, and builds a different machine by using it. On the other hand, instance selection, weighting, and ensemble learning [8, 9] are three most common measures taken to manage the effects concept drift. Building upon instance selection and involving generalization from a window are the most common ways to handle concept drift. These algorithms are called learning with multiple concept descriptions. Finally, Drift Detection Method (DDM) [10], and Early Drift Detection Method (EDDM) [11] are two typical concept drift algorithms in this research area. While DDM shows good behavior sudden change detection, it has difficulties when the changes happen slowly and gradually. On the other hand, EDDM improves the gradual changes detection. Later, the performance of DDM and EDDM are compared with our proposal.

III. Preprocessing Design

In this section, we present the design principles of DWM-CDD for spam mail filtering. This algorithm can detect both sudden and gradual changes quickly and accurately. Moreover, it speeds up the time needed for model construction. Figure 1 summarizes the preprocessing system require for this algorithm. The following preprocessing steps have been applied to DWM-CDD.

(1) Generally, e-mail consists of a header, and the body. Header is where the electronic address of the sender and the recipient(s) are indicated. The body contains all the information that the e-mail is composed for. The body of an e-mail may be in html or text format.

(2) E-mail Sort: First, e-mails are sorted by time assuming that the vocabulary of each e-mail is similar. Then, the subject of the e-mails from both the header and the body is selected as a text file.

(3) Tokenization: The words in the e-mails are separated from each other, and each word is considered as a token. In other words, words are extracted from the body of e-mails.

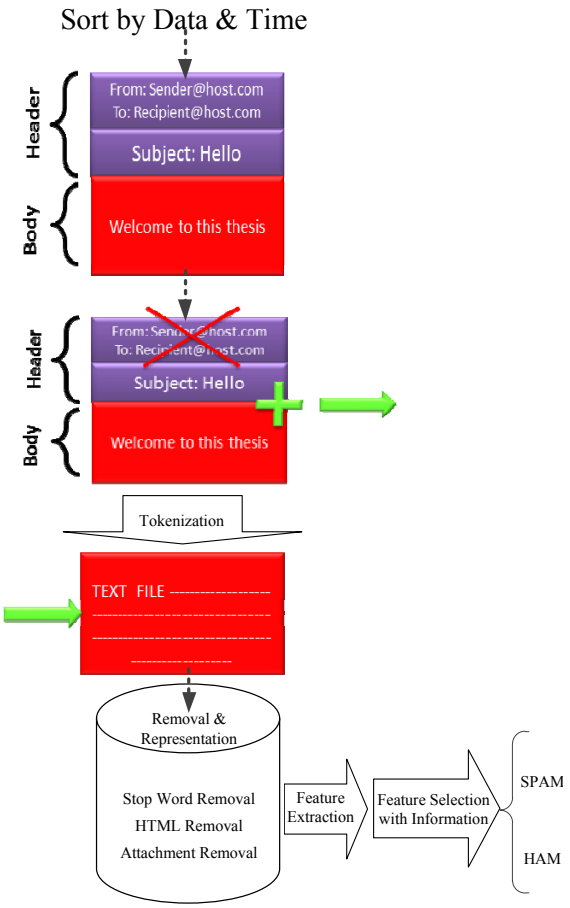

Fig. 1 Preprocessing system

(4) Stop Word Removal: Stop words such as "to", "a", and "for" are deleted. Stop words are the words which are used in all of the e-mails, and these are the key words which can separate ham e-mails from spam e-mails.
- First, database for stop words is built through the Internet and English books.
- By this procedure, some of the words which are not important for us are deleted, and consequently the number of words in each e-mail decreased.

(5) Html Removal: All the html tags are cleared. Next, attachments of e-mails are removed, and instead <attachment> is written.

(6) Stemming or Lemmatization: All words are returned to their root. For example the word "receiving" is changed to "receive". In order to reducing the number of words in e-mails, the words which are not important are deleted.

(7) Representation: The words are changed to the usable forms for algorithms. In other words, text classification is performed.
- The common approach to display text is based on vector spaces. The components of the vector show the weight of the feature. However, in this system, term frequency/ inverse document frequency (tf-idf) is used. This

method puts more emphasis on the features that are more frequent in the text. In addition, if this feature is repeated in several texts, lesser weight is given to that feature.

- After doing all the above steps, features of all e-mails are extracted, and are saved in feature vector.

(8) Feature Selection: The volume of feature vector is reduced in massive documents. In this step, not only the burden of processing for learning but also processing for classifying is reduced. In addition, the measure of efficiency is increased. Hence, feature selection has important role on reducing the dimension of vectors. From all the ways of feature selection, information gain is more effective and useful. Information gain has adaptable turnover in spam classification.

IV. Algorithm

Figure 2 illustrates the proposed algorithm for detecting concept drift. The algorithm has three different steps, control level, warning level, and alarm level. When there is no change in control level, the algorithm thinks about the changes that may happen in warning level. In addition, when the probability of changes increases, then, the algorithm is shifted to alarm level. Concept drift happens in this level.

For above purpose, DWM-CDD stores training examples in short-term memory when the algorithm is reached the warning level. Then, it rebuilds the online classifier from the stored examples if it has reached the drift level. This memory slightly improves their predictive accuracy immediately after the rebuild of the online classifier.

As shown in Fig. 1, the input data is the stream which is the set of vectors. Each vector is the instances with their features. First, a window size with the length of W is defined. These data are classified by the base algorithm, DWM. Then, the results feed to the proposed algorithm, CDD. Generally speaking, the proposed algorithm checks the data in the window to find how many of them have been classified correctly. S shows the number of data which has been classified correctly. Next, the algorithm compares them with the data which are not in the window. Finally, the algorithm compares these two rates by the statistical method. If the results are very different from each other, it shows that concept drift has happened.

The algorithm starts detecting drift after satisfying $n \geq 2W$. The sorted examples are removed when $P \geq a_W$. This algorithm uses two levels of significance which are a_W and a_d like EDDM. In addition, examples are stored in short term memory while $r / (n - W) > s / W$ and $P < a_d$, and then the classifier is built from the stored examples again. In addition, all the variables are reinitialized until $r/ (n - W) > s / W$ and $P < a_W$. When $P_A=P_B$ concept drift does not occur. Instead, it chooses P-value of the test for testing the new examples. The concept drift detection start working when $n = 2W$.

Parameters: W: window size, 1- a_d , 1- a_W: for significant levels
n: number of instances that classifier learned
r: number of correct classifications among W examples
W: the most recent examples

```
1.  0→ n, r, s, {w_t}, B ← □
2.  Online classifier: H: X → Y
3.  for each sample (x_t □ X,  y_t □ Y) do
4.     Output: H (x_t)
5.     increase n
6.     Set w_t ← [ H (x_t) = y_t]
7.          if w_t is true then
8.             increase s
9.        end if
10.      if  w_t-w is true then
11.        increase r; decrease s
12.     end if
13.      train online classifier H with (x_t, y_t )
14.      / / detecting concept drift//
15.      if  n ≥ 2W then
16.          P ← P of T(r, s, n- W, W)
17.       if r / (n - W) > s / W and  P <  a_d  then
18.         rebuild the classifier from B
19.           reinitialize n, r, s, { w_t }, B
20.          else  if r/ (n - W) > s / W and P < a_W  then
21.              add (x_t, y_t) to B
22.          else
23.            reinitialize B
       end if
     end if
   end for
```

Fig. 2 DWM-CDD algorithm

V. Evaluation and Experimental Results

In this section, we analyze and compare the performance of DWM-CDD with two typical algorithms EDDM and DDM. Tables I and II summarize our comparisons focusing on the following measures:

- Correctly classified: The percentage of spam e-mails recognized by classifier.
- Incorrectly classified: The percentage of spam e-mails recognized by classifier as ham e-mails.
- Kappa statistics: The consistency between predicted value and the true one.
- Recall: The fraction of all spam messages classified by the filter to be spam.
- Precision: The fraction of messages classified by the filter as spam that actually are spam.
- F-measure: The value of F measure can show how accurate the algorithm has been.
- ROC area: ROC curves provide a valuable insight into the tradeoff between ham and spam accuracy.

Table I shows the performance and the accuracy of DWM-CDD compared with two typical previously proposed algorithms. DWM-CDD can recognize 79% of spam e-mails by classifier correctly. Compared with EDDM and DDM, DWM-CDD improves the accuracy of those algorithms 23.5% and 26.4% respectively. In addition, DWM-CDD improves the other measurements compared with EDDM and DDM. The results show that DWM-CDD can recognize and classify spam e-mails from ham ones more accurately.

Table II shows the required time to construct the model between DWM-CDD, EDDM and DDM. For this purpose, we

have experimented with two open source software, TREC and Spam-Assassin, designed form spam e-mail filtering. Starting with TREC, we prepared 92000 e-mails containing 4000 ham e-mails and 52000 spam e-mails. In addition, with Spam-Assassin we prepared 6047 e-mails containing 1897 spam e-mails and 4150 ham e-mails. Finally, the result of comparisons with two software shows that DWM-CDD needs less time to construct the model compare with EDDM and DDM.

TABLE I
COMPARISON OF DWM-CDD WITH EDDM AND DDM

Statistical Feature	EDDM	DDM	DWM-CDD
Correctly Classified	55.4%	52.6%	79.0%
Incorrectly Classified	44.5%	47.3%	20.9%
Kappa Statistics	0.046	0.04	0.38
Recall	0.55	0.51	0.79
Precision	0.53	0.50	0.77
F-measure	0.54	0.52	0.77
ROC Area	0.50	0.50	0.76

TABLE II
THE TIME REQUIRED FOR MODEL CONSTRUCTION (SEC)

Algorithms	Spam-Assassin	TREC
DWM-CDD	1	6.88
EDDM	4.6	8.94
DDM	6	10.12

VI. CONCLUSION

In this paper, we introduced the algorithm of Dynamic Weighted Majority Concept Drift Detection (DWM-CDD). This algorithm is for concept drift in content-based filtering category.

At the end, we conclude our proposal with some of its contributions:

- DWM-CDD can detect both gradual and sudden changes.
- DWM-CDD can recognize spam e-mails more accurately compared with the other ones.
- DWM-CDD improves the parameters of spam e-mail filter, and covers 100% of cases compared with the other algorithm.
- The time required to construct the model is fast compared with previously proposed algorithms.

REFERENCES

[1] An Osterman Research White Paper "The Advantages of Using Traffic-Shaping Techniques to Control Spam," *Osterman Research, Inc*., pp. 1-6, Jan. 2007.

[2] T. S. Guzella, and W. M. Caminhas, "A Review of Machine Learning Approaches to Spam Filtering," *Elsevier, Expert Systems with Applications*, vol. 36, no. 7, pp. 10206-10222, 2009.

[3] A. Ciltik, and T. Gungor, "Time-Efficient Spam E-mail Filtering using n-Gram Models," *Pattern Recognition Letters*, vol. 29, no. 1, pp. 19-33, Jan. 2008.

[4] E. Blanzieri, and A. Bryl, "A Survey of Learning-based Techniques of Email Spam Filtering," *Artificial Intelligence Review*, vol. 29, no.1, pp. 63-922008

[5] I. Zliobate, "Learning under Concept Drift: an Overview," *Technical Report on Artificial Intelligence,* Vilinios University, pp. 371–391, 2010.

[6] Q. Zhu, X. Hu, Y. Zhang, and P. Li, "A Double-Window-based Classification Algorithm for Concept Drifting Data Streams," *proceedings of IEEE International Conference on Granular Computing (GrC)*, CA, USA, 2010, pp. 639-644.

[7] Z. Ouyang, and M. Zou, "Mining Concept-Drifting and Noisy Data Streams using Ensemble Classifiers," *proceedings of IEEE International Conference on Artificial Intelligence and Computational Intelligence (AICI 2009)*, Shanghai, China, 2009, pp. 360-364.

[8] A. Tsymbal, "The Problem of Concept Drift: Definitions and Related Work," *Technical report TCD-CS-2004-15,* Trinity College Dublin, Ireland, pp.123-. 130, 2004.

[9] J.Z. Kolter, and M.A. Maloof, "Dynamic Weighted Majority: A New Ensemble Method for Tracking Concept Drift," *Proceedings of IEEE Third International Conference on Data Mining*, Washington DC, USA, 2003, pp. 123-130.

[10] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with Drift Detection," *Lecture Notes in Computer Science*, vol. 3171/2204, pp. 66-112, 2004.

[11] M.B. Jose, J.D.C. Avila, R. Fidalgo, A. Bifet, R. Gavalda, and R.M. Bueno, "Early Drift Detection Method," *Fourth International Workshop on Knowledge Discovery from Data Streams*, Berlin, Germany, 2006, pp. 77-86.