JMLR: Workshop and Conference Proceedings 17 (2011) 12–18 2nd Workshop on Applications of Pattern Analysis

Using GNUsmail to Compare Data Stream Mining Methods for On-line Email Classification

José M. Carmona-Cejudo JMCARMONA@LCC.UMA.ES
Manuel Baena-García MBAENA@LCC.UMA.ES
José del Campo-Ávila JCAMPO@LCC.UMA.ES
Rafael Morales-Bueno MORALES@LCC.UMA.ES
Universidad de Málaga, Spain

João Gama JGAMA@FEP.UP.PT
University of Porto, Portugal

Albert Bifet ABIFET@CS.WAIKATO.AC.NZ
University of Waikato, New Zealand

Editor: Tom Diethe, José L. Balcázar, John Shawe-Taylor, and Cristina Tîrnăucă

Abstract

Real-time classification of emails is a challenging task because of its online nature, and also because email streams are subject to concept drift. Identifying email spam, where only two different labels or classes are defined (spam or not spam), has received great attention in the literature. We are nevertheless interested in a more specific classification where multiple folders exist, which is an additional source of complexity: the class can have a very large number of different values. Moreover, neither cross-validation nor other sampling procedures are suitable for evaluation in data stream contexts, which is why other metrics, like the prequential error, have been proposed. In this paper, we present GNUsmail, an open-source extensible framework for email classification, and we focus on its ability to perform online evaluation. GNUsmails architecture supports incremental and online learning, and it can be used to compare different data stream mining methods, using state-of-art online evaluation metrics. Besides describing the framework, characterized by two overlapping phases, we show how it can be used to compare different algorithms in order to find the most appropriate one. The GNUsmail source code includes a tool for launching replicable experiments.

1. Introduction

Nowadays emails can be easily stored due to the low cost of data storage devices. It is common to file emails with a certain level of interest, group them into some hierarchical structure using category-based folders, or tag them with different labels. However, email classification can pose more difficulties than spam detection because of the potentially high number of possible folders. Traditionally, classification tasks had to be manually carried out by the user, which was very time-consuming. To help with these tedious tasks, most email clients support hand-crafted rules that can automate them attending to different predefined characteristics (sender, subject, nature of content, etc.).

Two different approaches have been defined, depending on how the messages are processed: *batch learning*, where the whole dataset is available before the beginning of the

learning process, and *online learning*, where data are continually being received over time. Different solutions have been proposed for each approach. Irrespective of the type of approach, there is a lack of systems to compare and evaluate different machine learning models for email classification. GNUsmail (Carmona-Cejudo et al., 2010) is a framework that allows to compare different email classification algorithms.

The aim of this paper is thus to introduce new improvements to GNUsmail and show its qualities as a platform to carry out *replicable experimentation* in the field of data stream mining applied to email classification. We analyze email as a stream of data with great amounts of messages which arrive continuously and must be processed only once (*incremental learning*). In this online approach, we consider that the nature of emails within a folder may change over time. We review and implement different proposals to evaluate and compare data stream mining methods, including modern approaches such as sliding windows and fading factors (Gama, 2010) applied to the classical prequential error. Furthermore, we adapt GNUsmail to use recently proposed statistical tests to detect significant differences between the performance of online algorithms, and then employ the framework to evaluate different machine learning methods.

2. GNUsmail: Architecture and Characteristics

GNUsmail (Carmona-Cejudo et al., 2010) is an open-source framework for online adaptive email classification, with an extensible *text preprocessing module*, based on the concept of filters that extract attributes from emails, and an equally extensible *learning module* into which new algorithms, methods and libraries can be easily integrated. GNUsmail contains configurable modules for reading email, preprocessing text and learning. In the learning process, the email messages are read as the model is built, because, for practical reasons, email messages are analyzed as an infinite flow of data.

The source code is available at `http://code.google.com/p/gnusmail/` and it is licensed under the GPLv3 license.

We now explain in more detail the different modules integrated in GNUsmail.

2.1. Reading Email and Text Preprocessing Module

The *reading email module* can obtain email messages from different data sources, such as a local filesystem or a remote IMAP server. This allows it to process datasets like ENRON or to process personal email messages from remote servers like Gmail.

The main feature of the *text preprocessing module* is a multi-layer filter structure, responsible for performing feature extraction tasks. The `Inbox` and `Sent` folders are skipped in the learning process because they can be thought of as 'non-specific' folders. Every mail belonging to any other folder (that is, to any *topical folder*) goes through a pipeline of linguistic operators which extract relevant features from it (Sebastiani, 2002). As the number of possible features is prohibitively large, only the most relevant ones are selected, so GNUsmail performs a feature selection process using different methods for feature selection (Forman, 2003; Manning and Schütze, 2003). Some ready-to-use filters are implemented as part of the GNUsmail core, and new ones can be incorporated, giving developers the chance to implement their own filters, as well as test and evaluate different techniques. We have

implemented a linguistic filter to extract attributes based on relevant words. It is based on the ranking provided by the folder-wise *tf-idf* function. We have implemented several filters to extract non-linguistic features such as *CC*, *BCC*, *sender*, *number of receivers*, *domain of sender*, *size*, *number of attachments* or *body/subject length*. We have also implemented filters to extract metalinguistic features, such as *capital letters proportion*, or the *language* the messsage is written in.

2.2. Learning Module

The *learning module* allows the incorporation of a wide variety of stream-based algorithms, such as those included in the WEKA (Hall et al., 2009) or MOA (Bifet et al., 2009, 2010) frameworks. WEKA methods are used mainly with small datasets in environments without time and memory restrictions. MOA is a data mining framework that scales to more demanding problems, since it is designed for mining data streams.

GNUsmail offers three updateable classifiers from the WEKA framework, more can be easily added though. The first one is Multinomial Naïve Bayes, a probabilistic classifier which is commonly used as a baseline for text classification. The other two classifiers belong to the family of lazy classifiers, which store all or a subset of the learning examples; when a new sample is given as input to the classifier, a subset of similar stored examples is used to provide the desired classification. IBk is one of these methods, a k-nearest neighbours algorithm, to be precise, that averages the k nearest neighbours to provide the final classification for a given input. NN-ge (Martin, 1995) is another nearest-neighbours-like algorithm that groups together examples within the same class. This reduces the number of classification errors that result from inaccuracies of the distance function.

In the streaming scenario, GNUsmail uses MOA by including its tools for evaluation, a collection of classifier algorithms for evolving data streams, some ensemble methods, and drift detection methods. HoeffdingTree (VFDT) is a classifier algorithm developed by Domingos et al. (Domingos and Hulten, 2000) to efficiently construct decision trees in stream-oriented environments. An appealing feature of Hoeffding trees is that they offer sound guarantees of performance. They can approach the accuracy of non-streaming decision trees with an unlimited number of training examples. An improved learner available in MOA is the `HoeffdingTreeNB`, which is a Hoeffding Tree with Naïve Bayes classifiers at leaves, and a more accurate one is the `HoeffdingTreeNBAdaptive`, which monitors the error rate of majority class and Naïve Bayes decisions in every leaf, and chooses to employ Naïve Bayes decisions only where they proved accurate in the past. Additionally, some ensemble methods from the MOA framework are included, such as `OzaBag` and `OzaBoost`; incremental online bagging and boosting methods respectively. For concept drift detection, we have included DDM (Gama et al., 2004) that detects change in the accuracy error of the classifier by comparing window statistics.

3. Evaluation of Data Stream Mining Methods

In data stream contexts, neither cross-validation nor other sampling procedures are suitable for evaluation. Other methods, like the *prequential* measure, are more appropriate (Gama, 2010). By using prequential methods, a prediction is made for each new example (i-th

example) using the current model. Once the real class is known, we can compute the loss function ($L(y_i, \hat{y}_i)$, where L is a loss function – for example $0-1$ loss function –, y_i is the real class and $\hat{y}_i$ is the prediction), and a cumulative loss function (S_i) is updated: $S_i = \sum_{j=1}^{i} L(y_j, \hat{y}_j)$. After that, the model is updated using that example. Thus, we can evaluate the performance of the system (that is influenced by the increasing number of examples that are constantly arriving) and the possible evolution of the models (because of concept drift). Using the previous cumulative loss function (S_i), we can estimate the mean loss at every moment (i): $E_i = S_i/i$.

Although the prequential approach is a pessimistic estimator when compared with the holdout approach (where an independent test set is previously reserved), it can perform similarly to holdout by using forgetting mechanisms, like sliding windows or fading factors, and overcome other problems that are present in the holdout approach (Gama, 2010). The method based on sliding window is about considering only the last examples. Therefore, not all the examples are used to evaluate the performance: the former are forgotten and the most recent ones are stored in a window of size w. Those w examples in the window are the examples used to calculate different performance measures. On the other hand, using fading factors, which is the preferred method according to (Gama, 2010), allows to decrease the influence of the examples in the measure as they get older. For example, if we compute the loss function (L) for every single example, the prequential error at moment i using fading factors is defined as $E_i = S_i/B_i$, where $S_i = L(y_j, \hat{y}_j) + \alpha \cdot S_{i-1}$ and $B_i = 1 + \alpha \cdot B_{i-1}$ ($\alpha < 1$).

There exist different ways of comparing the performance of two classifiers; McNemar test being one of the most used tests. This test needs to store and update two quantities: a (the quantity of the examples misclassified by the first classifier and not by the second one) and b (the quantity of the examples misclassified by the second classifier and not by the first one). The McNemar statistic (M) rejects the null hypothesis (the performances are the same) with confidence level of 0.99 if its value is greater than 6.635, since it follows a χ^2 distribution. It is calculated as $M = sign(a-b) \times \frac{(a-b)^2}{a+b}$.

In order to extend its usage to the prequential approach we have two options. If we consider a sliding window context, we only need to use the examples in the window to compute the statistic. But if we consider the usage of fading factors, we should adapt the definition in the following way:

$$M_i = sign(a_i - b_i) \times \frac{(a_i - b_i)^2}{a_i + b_i}$$

where $a_i = f_i + \alpha \cdot a_{i-1}$ and $b_i = g_i + \alpha \cdot b_{i-1}$, being $f_i = 1$ if and only if the i-th example is misclassified by the first classifier and not by the second one ($f_i = 0$ otherwise) and $g_i = 1$ if and only if the i-th example is misclassified by the second classifier and not by the first one ($g_i = 0$ otherwise). Note that it still follows a χ^2 distribution.

4. Replicable Experimentation

In this section, we describe how we have adapted GNUsmail to include the exposed evaluation methods. We also explain the experimental setup that we have used to test the

proposed adaptation, as well as its results. All experiments are replicable since GNUsmail incorporates such functionality.

4.1. Experimental Setup

We have evaluated our framework for both incremental and online learning schemes, using the ENRON email dataset (Klimt and Yang, 2004). Following the approach of (Bekkerman et al., 2004) and (Bermejo et al., 2008), we have used seven specific users (*beck-s*, *farmer-d*, *kaminski-v*, *kitchen-l*, *lokay-m*, *sanders-r* and *williams-w3*), based on their interest in terms of number of messages and folders. For each of these users, we have used only topic folders with more than two messages. The GNUsmail experimentation launcher first checks whether the ENRON dataset is already available, offering to download it if necessary.

As classification attributes we have used the number of recipients, the sender (broken down into username and domain), the body length, the capital letters proportion, the size of email, the subject length and the most relevant words, as we explained in Subsection 2.1.

For both the incremental and the online approach, the messages are analyzed in chronological order, and they are given to the underlying learning algorithms one by one. In this way, the system simulates how the model would work when classifying incoming emails in the real world. The specific algorithms to be used and compared are configurable. For each author, we compare the online performance of the following algorithms:

- OzaBag over NNge, using DDM for concept drift detection
- NNge
- Hoeffding Trees
- Majority class

Instead of using the classical prequential error, the launcher can be configured to use sliding windows or fading factors. For each algorithm, GNUsmail plots the prequential-based metrics, to visually analyse the differences in performance. These plots also show the concept drifts when there is a concept drift detector associated with some algorithm.

In addition, for the sake of performing some statistical tests, GNUsmail also includes the McNemar test because it can be adapted to deal with the online setting (Gama, 2010).

4.2. Results

In Table 4.1 we show the final MOA prequential accuracies with bagging of DDM and NN-ge for each folder. As can be see in this table, the classification results are similar to those in (Bekkerman et al., 2004) and (Bermejo et al., 2008). The final results depend largely on each specific author, and, more precisely, on their folders.

Better results are achieved when using fading factors, since the effect of poor initial performance is reduced because of the limited number of available messages at the beginning.

NN-ge outperforms other basic methods, whose relative performance varies for every author. The performance obtained by NN-ge is comparable with the best results obtained by (Bekkerman et al., 2004) and (Bermejo et al., 2008), which use SVM-based approaches. As a consequence of using OzaBag and DDM over NNge, the results are further improved.

Folder	Correct/Total	Percentage	Folder	Correct/Total	Percentage
beck-s	1071/1941	55.18%	farmer-d	2743/3590	76.41%
europe	131/162	80.86%	logistics	1077/1177	91.58%
calendar	104/123	84.55%	tufco	588/608	96.71%
recruiting	89/114	78.07%	wellhead	210/337	62.32%
doorstep	49/86	56.97%	personal	211/320	65.94%
kaminsky-v	1798/2699	66.62%	kitchen-l	2254/3790	59.47%
universities	298/365	81.64%	esvl	640/712	89.89%
resumes	420/545	77.06%	hr	159/299	53.18%
personal	154/278	55.4%	east_power	160/253	63.24%
conferences	163/221	73.76%	regulatory	210/242	86.78%
lokay-m	1953/2479	78.78%	sanders-r	887/1207	73.49%
tw_commercial_group	1095/1156	94.72%	iso_pricecaps	404/420	96.19%
corporate	345/400	86.25%	nsm	109/119	91.6%
articles	152/232	65.51%	senator_dunn	43/83	51.81%
enron_t_s	86/176	48.86%	px	49/68	72.06%
williams-w3	2653/2778	95.5%			
schedule_crawler	1397/1398	99.91%			
bill_williams_iii	1000/1021	97.94%			
hr	74/86	86.05%			
symsees	74/81	91.36%			

The methods based on HoeffdingTree have lower accuracy than the methods based on NNge when directly applied to the domain of email classification. These streaming decision tree methods frequently need millions of examples to start showing good performance. Within the email classification domain, it is not unusual to be dealing with thousands of messages, but millions of messages would be something uncommon.

5. Conclusion

In this paper, we have presented different methods to evaluate data stream mining algorithm in the domain of email classification. We have improved GNUsmail, an open-source extensible framework for email classification, which incorporates a flexible architecture into which new feature extraction, feature selection, learning and evaluation methods can be incorporated. In this enhanced version of the framework, we incorporate recently proposed evaluation methods for online learning with concept drift. Such evaluation methods improve the prequential error measures by using mechanisms to reduce the effect of past examples, such as sliding windows and fading factors.

Current online learning algorithm implementations have an important limitation that affects the learning process: learning attributes have to be fixed before beginning the induction of the algorithm. They need to know all the attributes, values and classes before the learning itself, since it is not possible to start using a new attribute in the middle of the lifetime of a learning model. Future methods should support online addition of new features.

Acknowledgments

This work has been partially supported by the SESAAME project (code TIN2008-06582-C03-03) of the MICINN, Spain. José M. Carmona-Cejudo is supported by AP2009-1457grant from the Spanish government.

References

R. Bekkerman, A. Mccallum, and G. Huang. Automatic categorization of email into folders: Benchmark experiments on Enron and SRI Corpora. Technical report, Center for Intelligent Information Retrieval, 2004.

P. Bermejo, J. A. Gámez, J. M. Puerta, and R. Uribe-Paredes. Improving KNN-based e-mail classification into folders generating class-balanced datasets. In *IPMU-2008*, pages 529–536, 2008.

A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà. New ensemble methods for evolving data streams. In *KDD-2009*, pages 139–148, 2009.

Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Philipp Kranen, Hardy Kremer, Timm Jansen, and Thomas Seidl. MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering. *JMLR - Proceedings Track*, 11:44–50, 2010.

José M. Carmona-Cejudo, Manuel Baena-García, José del Campo-Ávila, Rafael Morales Bueno, and Albert Bifet. Gnusmail: Open framework for on-line email classification. In *ECAI*, pages 1141–1142, 2010.

Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In *Knowledge Discovery and Data Mining*, pages 71–80, 2000.

G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

J. Gama. *Knowledge Discovery from Data Streams*. CRC Press, 2010.

J. Gama, P. Medas, G. Castillo, and P. Rodrigues. Learning with drift detection. In *SBIA Brazilian Symposium on Artificial Intelligence*, pages 286–295, 2004.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. In *SIGKDD Explorations*, volume 11-1, pages 10–18, 2009.

B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In *Proceedings of the 15th European Conference on Machine Learning (ECML-2004)*, 2004.

C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 2003.

Brent Martin. Instance-based learning: Nearest neighbour with generalization. Master's thesis, University of Waikato, 1995.

F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47, 2002.