Chronological Sampling for Email Filtering

Ching-Lung Fu[2], Daniel Silver[1⋆], and James Blustein[2]

[1] Acadia University, Wolfville, Nova Scotia, Canada
[2] Dalhousie University, Halifax, Nova Scotia, Canada

Abstract. User models for email filtering should be developed from appropriate training and test sets. A k-fold cross-validation is commonly presented in the literature as a method of mixing old and new messages to produce these data sets. We show that this results in overly optimistic estimates of the email filter's accuracy in classifying future messages because the test set has a higher probability of containing messages that are similar to those in the training set. We propose the k-fold chronological cross-validation method that preserves the chronology of the email messages in the test set.

1 Introduction

Our research into spam email filtering began with an investigation of existing filtering systems that are based on learned users models. Often, we found the reported accuracy of these systems to be overly optimistic [1]. We found it difficult to create models with the same high level of accuracy as published when using the same or independent datasets. Other authors have made similar observations [2]. Although much of the difference between the earlier evaluations and ours can be attributed to differences in the mix of legitimate and spam emails in the datasets, we speculated that another important factor is the method of testing that is employed.

Our work with machine learning models for spam filtering has shown that time is an important factor for valid testing; i.e. the order of incoming email messages must be preserved so as to properly test a model. Unfortunately, many results reported in the literature are based on a k-fold cross-validation methodology that does not preserve the temporal nature of email messages. It is common practice with cross-validation to mix the available data prior to sampling training and test sets so as to ensure a fair distribution of legitimate and spam messages [3]. However that practice, by mixing older messages with more recent ones, generates training sets that unfairly contain future messages. The mixing ignores an important dynamic feature of spam e-mail, namely that the generator (spammer) is an adversary who incorporates information about filtering techniques into their next generation spam [4].

If the temporal aspect is not considered, the performance of the model on future predictions may be significantly less than that estimated. A fair test set should contain only messages received after those used to develop the model. Commonly used data sets available on the web, such as the Ling-Spam corpus [5], do not even contain a time stamp in the data. We present a comparison of a spam filtering system tested using cross-validation and a temporal preserving approach.

⋆ Corresponding author (danny.silver@acadiau.ca)

2 Background

The k-fold cross-validation method is a standard method for comparing different models [3]. In k-fold cross-validation the dataset is divided into k subsets of approximately equal size. Model generation and testing is repeated k times. Each time a different subset is selected as a test set and the remaining subsets are selected for training. In some machine learning algorithms (*e.g.* inductive decision trees and artificial neural networks) it is necessary to select a part of the training set as a validation set to reduce the likelihood of over fitting. Each subset can be in the test set exactly once and in the training set $(k-1)$ times. Before splitting the dataset, it is common to randomly sort all examples in the dataset, to ensure that they are evenly distributed, before creating the k subsets. The intention of cross-validation is that it will better estimate the true accuracy of the resulting models, based on the mean accuracy calculated over the k evaluations. In the addition, the standard deviation around the mean can be used to produce confidence intervals and to determine the statistical significance between different models, or machine learning algorithms.

Consider the effect of randomly mixing the legitimate and spam messages prior to undertaking a k-fold standard cross-validation (SCV). When the datasets are mixed, possible future examples are injected into the training set thereby providing the model with an unfair look at future features. Figure 1 illustrates the problem of mixing old and new examples under SCV: If **A**, **B**, and **C** are three main types of examples in the dataset, let **A** be the oldest and **C** the newest. In SCV, all three types of examples are evenly distributed in the training set, validation set, and test set. This distribution provides the best opportunity for a model to perform well on the test set. However, in reality, type **A** and **B** have the highest probability of being available in the data set during the time of model development. **C** may not have appeared until after **A** and **B**. Thus, the mixing of examples in SCV provides an unrealistic set of future examples in the training set. We claim that this is one of the major reasons why many of the reported results on spam filtering are overly optimistic.

3 Chronological Cross-Validation

We propose k-fold Chronological Cross-Validation (CCV) as a more realistic evaluation method of data for any temporally-sensitive model (including e-mail) that selects the training and test sets while the data is in chronological order. The test set will then simulate the classification of future messages based on a model produced from prior messages and, therefore, the test set accuracy will better estimate the true accuracy of the email filter. CCV maintains the chronology of the email messages as the evaluation process moves along the chronological order of the data set. A ten-fold CCV is depicted in Fig. 2. We propose that this new cross-validation approach will reduce the probability of over-estimating the effectiveness of the model. CCV method is as follows:

1. The data set is sorted chronologically;
2. The data set is divided into $2k-1$ blocks;
3. k folds (or repetitions) are undertaken as follows:

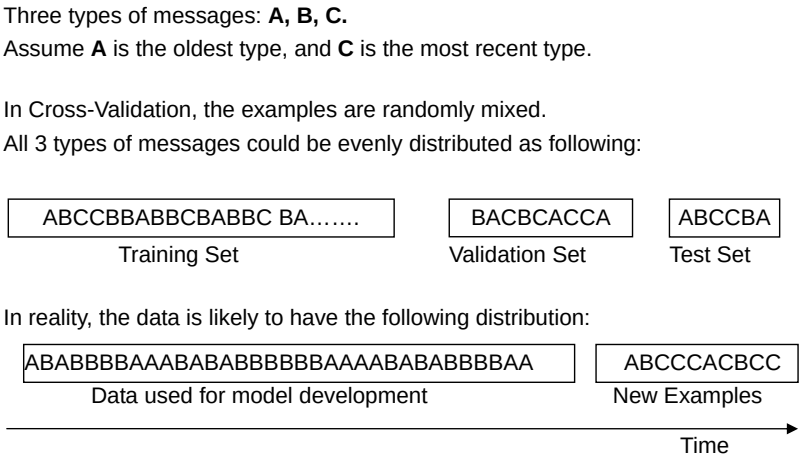

Fig. 1. A problem with the random mix of examples in standard Cross-Validation.

(a) In the first repetition, blocks 1 to k are selected for evaluation;
(b) The first $k-(v+1)$ blocks are selected as the training set, the following v blocks are selected as validation set, and the last block is reserved for the test set;
(c) Each later fold advances one data block in chronological order and the oldest data block is abandoned (for example in Fig. 2, in the second repetition, block 1 is abandoned and blocks 2 to $k+1$ are selected for evaluation);

4. The procedure is repeated k times, until data block $2k-1$ has been tested.

4 Empirical Studies

Two studies were undertaken using one email dataset called AcadiaSpam, collected from a single individual from January to May 2003, working at Acadia University. The set consisted of 1454 spam messages and 1431 legitimate messages. The initial study was conducted with a subset of these emails using one experimental design and the second was conducted with all of the data using a slightly different design.

4.1 Experiment 1

The initial study was undertaken during the development of a prototype intelligent email client. A small subset of the data was chosen so as to quickly determine if the proposal had merit for larger scale testing. The objective was to determine the severity of SCV over-estimates the true accuracy of hypotheses as compared to CCV.

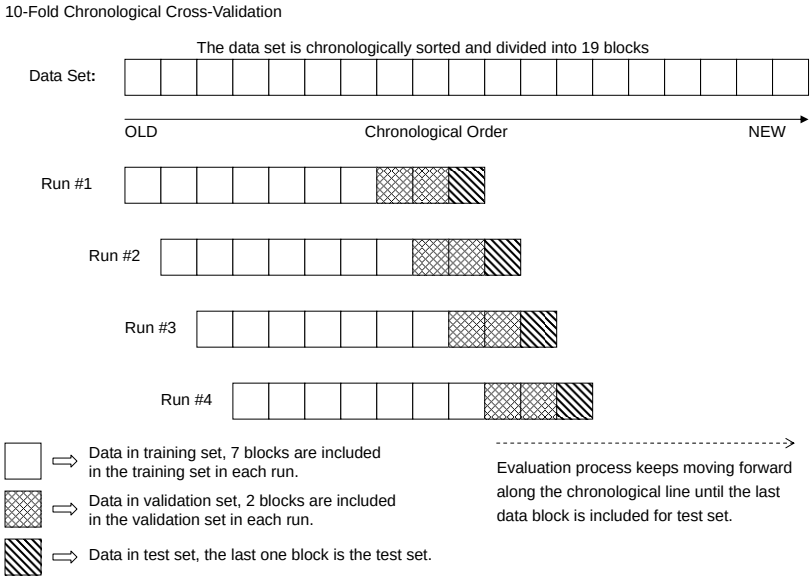

Fig. 2. A 10-fold Chronological Cross-Validation.

Method. The first 500 legitimate messages and 500 spam messages were selected from the AcadiaSpam dataset and stored in chronological order. A $k = 6$ was chosen for this initial experiment; therefore, 6 models were repeatedly developed and tested against their respective test sets under each method.

A block of 500 messages was chosen for each repetition starting at the earliest email message. On each repetition, the block was moved 100 messages forward in the chronology. From each block of messages, 300 were selected for a training set, 100 for a tuning set, and 100 for a test set. Two data sampling methods were used to create the sets. For the SCV method, the message data for the three sets were randomly chosen. For the CCV method, the most recent messages were selected for the test set and the remaining messages randomly distributed to the training and tuning sets. The tuning set was used to prevent over-fitting of the neural network to the training set.

Prior to model development, 200 features were extracted from each message using traditional information retrieval methods (removing stop words, performing word stemming, and collecting word statistics). A standard back-propagation neural network with a momentum term was used to develop the spam filter models [3]. The network had 200 inputs, 20 hidden nodes and 1 output node. A learning rate of 0.1 and a momentum factor of 0.8 were used to train the networks to a maximum of $10,000$ iterations.

Since the output of the network ranges from 0 to 1, messages with output greater than 0.5 were classified as legitimate, and all others were classified as spam. The models were evaluated based on their accuracy of classification, precision and recall of spam email messages. The calculations of accuracy, recall and precision follow [2].

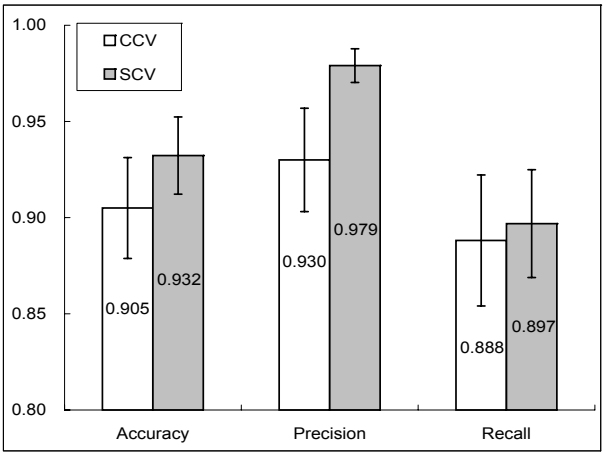

Fig. 3. Comparison of SCV and CCV on the 1000 AcadiaSpam dataset with $k = 6$.

Results and Discussion. Figure 3 shows that the SCV method estimates the true accuracy of the models to be 0.93. That figure is, on average, 2.5% higher than the CCV method's estimate ($p = 0.238$, based on a paired t-test). Similarly, the SCV method consistently produces the higher precision and recall models. The difference in the precision values is most significant at 5% ($p = 0.0249$). Our conjecture is that the SCV method unrealistically allows the modelling system to identify features of future spam emails. The SCV method over-estimates its performance on the test messages because the training set has a higher probability of containing messages that are similar to those in the test set. The CCV method generates a less accurate but more realistic model because the testing procedure simulates the classification of future incoming messages.

A potential flaw in this preliminary study is that it does not use a standard cross-validation method, as not all data was used in every repetition of SCV. This was done to keep the number of examples used by the two systems the same during each repetition. Although we suspect that a more standard SCV would further increase the performance gap between SCV and CCV we undertook a second study to investigate this potential concern about the validity of our results.

4.2 Experiment 2

The second study used all of the available data in a more traditional SCV approach in which all data is used in every repetition. As in the initial study, the objective was to show that SCV over-estimates the true accuracy of hypotheses as compared to CCV.

Method All messages in the AcadiaSpam dataset were used in this experiment (1454 spam messages and 1431 legitimate messages). For SCV, the messages were randomly ordered and divided into $k = 10$ blocks each consisting of 143 legitimate and 145 spam messages. Each repetition used 7 blocks as the training set, two blocks for a validation set, 1 block as a test set.

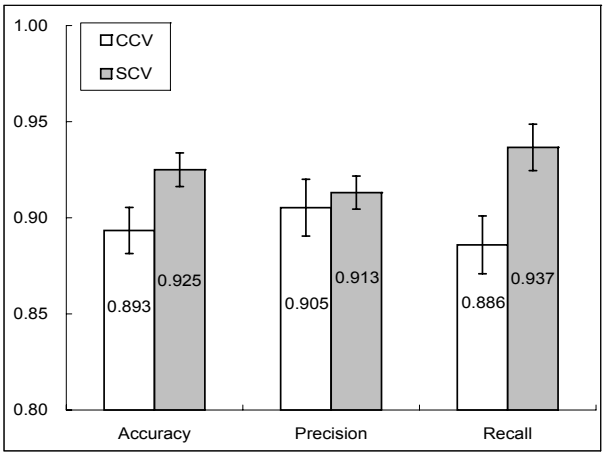

Fig. 4. Comparison of SCV and CCV on the 2880 AcadiaSpam dataset with $k = 10$.

For CCV, the AcadiaSpam dataset was chronologically sorted. Legitimate messages and spam messages were divided into 19 ($2k-1$, where $k = 10$) blocks. Each block has 151 messages consisting of approximately 75 legitimate messages and 76 spam messages. Each repetition of the experiment used a window of $k = 10$ blocks of messages starting with the oldest block. From these 10 blocks, the 7 oldest blocks are used as the training set, the next 2 blocks for the validation set, and the most recent block for the test set. For each new repetition of the experiment, the oldest block was removed from the window of 10 blocks and the next chronological block was added. Note that every message was used by both methods, however fewer examples were used in each repetition by CCV than by SCV. All other aspects of the method were the same as in the first experiment.

Results and Discussion. The results of this larger experiment, shown in Fig. 4, support the findings of the initial experiment. The SCV method produces hypotheses with superior performance in all 3 measurements (accuracy, recall, precision) as compared with CCV. The difference in mean accuracy between the two methods was found to be 3.1% ($p = 0.00057$, based on a paired t-test) up from 2.5% in the initial study. As in the initial study, the SCV method produces the highest precision and recall statistics. In this case, no significant difference was detected in the precision statistics ($p = 0.38$) but the recall statistics differed substantially ($p = 0.000067$).

Although the difference in the statistics for these methods could be attributed solely to the smaller training sets used to develop the CCV models in this study, the results of the initial study do not support that conclusion. When both methods used the same size training sets, the SCV method was still found to over-estimate the model's performance. We conclude that the difference in the statistics is caused by unrealistic mix of old and new examples in the training sets used to develop the SCV models.

5 Related Work

We have recently discovered work by Crawford *et al* [6] that agrees that the k-fold SCV method is unrealistic given the temporal nature of email. They describe a model development approach that accumulates the older messages in the training set and selects only the most recent data for the test set. This testing approach is in accord with how a real email filter must perform; therefore, it should provide a fair evaluation of the model's effectiveness. Beyond this the research emphasis and approach differs from our work. Crawford *et al* focus on model development over time whereas we are interested in a cross-validation method for estimating the true error of a model at any one point in time. Our CCV method purposely abandons older blocks of examples as it moves through its repetitions so as to better estimate model performance on a variety of examples.

6 Summary and Conclusion

We have considered the importance of maintaining the temporal nature of incoming email messages when developing a user model for email filtering. Although the k-fold SCV is commonly presented in the literature as a method of randomly mixing examples to produce training and test datasets, we have demonstrated that the method results in overly optimistic estimates of an email spam filter's accuracy in classifying future messages. Our conjecture is that the SCV method is inappropriate because it allows the modelling system to unfairly identify features of the test set spam emails. The SCV method over-estimates model performance on future messages because the training set has a higher probability of containing messages that are similar to those in the test set.

We propose the k-fold Chronological Cross-Validation (CCV) method as a step towards more realistic estimates of model performance. CCV generates less accurate but more realistic models because the testing procedure more properly simulates the classification of future messages. The CCV method can be used to more properly evaluate any complex user model that will change over time. Thus, it can better estimating a models ability to deal with *concept drift*: the change in a user model over time due to subtle changes in the user, their environment, or both [7]. More broadly, the CCV method can be applied to any learning task where the order of examples must be preserved.

The CCV method highlights the fact that more examples are needed to properly evaluate a user model when the preservation of example order is a requirement. The size of the time window, which depends on k, must be large enough to develop good models but small enough to allow sufficient blocks for cross-validation. Window size must also be sensitive to the mix of training examples and is likely to be different for each individual. These are a couple of the open problems that we would like to investigate in future research.

References

1. Clark, J., Koprinska, I., Poon, J.: A Neural Network Based Approach to Automated E-Mail Classification. In: Proceedings IEEE/WIC International Conference on Web Intelligence (WI2003), Halifax, Canada (2003) 562 – 569

2. Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C., Stamatopoulos, P.: Learning to Filter Spam E-Mail: A Comparison of a Naïve Bayesian and a Memory-Based Approach. In: Proc. of the Workshop on Machine Learning and Textual Information Access, 4th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD-2000), Lyon, France (2000) 1 – 13
3. Mitchell, T.: Machine Learning. McGraw Hill, New York, USA (1997)
4. Stern, H., Mason, J., Shepherd, M.: A linguistics-based attack on personalised statistical e-mail classifiers. Technical Report CS-2004-06, Dalhousie Univ. (2004)
5. Androutsopoulos, I.: Ling-spam corpus (2000) ⟨`http://www.aueb.gr/users/ion/data/lingspam_public.tar.gz`⟩.
6. Crawford, E., Kay, J., McCreath, E.: Automatic induction of rules for e-mail classification. In: The Sixth Australian Document Computing Symposium, Coffs Harbour, Australia (2001)
7. Widmer, G.: Combining Robustness and Flexibility in Learning Drifting Concepts. In: Proceedings of the 11th European Conference on Artifcial Intelligence (ECAI-94), Wiley, Chichester, UK (1994) 468–472