

A Comparison of Ensemble and Case-Base Maintenance Techniques for Handling Concept Drift in Spam Filtering

Sarah Jane Delany
Dublin Institute of Technology
Kevin Street, Dublin 8
sarahjane.delany@comp.dit.ie

Pádraig Cunningham and Alexey Tsymbal
Trinity College Dublin
Dublin 2
{padraig.cunningham, alexey.tsymbal}@cs.tcd.ie

Abstract

The problem of concept drift has recently received considerable attention in machine learning research. One important practical problem where concept drift needs to be addressed is spam filtering. The literature on concept drift shows that among the most promising approaches are ensembles and a variety of techniques for ensemble construction has been proposed. In this paper we compare the ensemble approach to an alternative lazy learning approach to concept drift whereby a single case-based classifier for spam filtering keeps itself up-to-date through a case-base maintenance protocol. The case-base maintenance approach offers a more straightforward strategy for handling concept drift than updating ensembles with new classifiers. We present an evaluation that shows that the case-base maintenance approach is as least as effective as/appears marginally more effective than a selection of ensemble techniques. The evaluation is complicated by the overriding importance of False Positives (FPs) in spam filtering. The ensemble approaches can have very good performance on FPs because it is possible to bias an ensemble more strongly away from FPs than it is to bias the single classifier. However this comes at considerable cost to the overall accuracy.

Introduction

While much of the research on machine learning has focused on static problems (Vapnik 1999), a significant issue in many real-world problems is a changing environment (Kelly, Hand, & Adams 1999). There are a variety of ways in which an environment may change, here we consider *concept drift*, where the underlying concept changes over time. We are specifically concerned with spam (i.e. junk email) filtering where the underlying concept being tracked changes over time. Sub-categories of legitimate email and spam will change over time as will the underlying distributions of these sub-categories. Concept drift in spam is particularly difficult as the spammers actively change the nature of their messages to elude spam filters.

Research has proposed a number of approaches to handling concept drift. Our earlier work on concept drift in spam has shown that an instance selection approach that

uses a case-base maintenance protocol with a single case-based classifier is effective (Delany *et al.* 2005). Other research on concept drift in general shows that ensemble approaches are among the most effective (Kolter & Maloof 2003; Kuncheva 2004; Stanley 2003; Street & Kim 2001; Wang *et al.* 2003). Kuncheva (2004) presents the ensemble approach to learning in changing environments as online learning with forgetting. The online learning is achieved by adding new members trained with the most recent data to the ensemble. And forgetting is achieved by deleting old or less-useful ensemble members. In this paper we take three variants of this idea and compare them with our case-base maintenance approach. The case-base maintenance protocol involves an initial case-base editing stage and then a case-base update procedure which regularly adds in new emails that are misclassified by the case-base. It also periodically performs a feature reselection process to ensure that the new features are reflected in the case representation.

In order to separate effects due to the ensembles from effects due to concept drift the evaluation is done in two stages. The first stage is a static or batch evaluation where the ensemble approaches are compared with the single case-base approach using cross-validation. This evaluation showed that the ensembles that were considered did not improve on the classification accuracy of the single case-base classifier in this domain. However, it did show that the ensembles could be configured to produce less False Positives (FPs, which are legitimate emails incorrectly classified as spam) than the single case-base. This is because there is more scope to bias the ‘decision-making’ of the ensemble.

The second stage is a dynamic or online evaluation which compares the performance of the ensembles with that of a single case-base classifier as the classifiers are incrementally updated at regular intervals with new examples of training data. We show from the performance of the baseline classifiers that there is considerable concept drift in this data. The dynamic evaluation shows that the single classifier incorporating the case update protocol is at least as good as the ensemble at tracking this concept drift. The evaluation also shows that an ensemble pruning policy whereby the best ensemble members are selected based on an assessment of their performance on recent data is the most effective of the ensemble approaches.

The body of this paper begins with a review of techniques

for handling concept drift. Then the approaches for handling concept drift in spam that are evaluated in this paper are described and evaluated and the paper concludes with a summary and suggestions for future work.

Techniques for Handling Concept Drift

An analysis of the machine learning literature on concept drift suggests that there are three general approaches; instance selection, instance weighting and ensemble learning. In instance selection the goal is to select instances that are relevant to the current concept. The most common concept drift technique is based on instance selection and involves generalising from a *window* that moves over recently arrived instances and uses the learnt concepts for prediction in the immediate future. Examples of window-based algorithms include the FLORA family of algorithms (Widmer & Kubat 1996), FRANN (Kubat & Widmer 1995) and Time-Windowed Forgetting (Salganicoff 1997). Some algorithms use a window of fixed size while others use heuristics to adjust the window size to the current extent of concept drift, e.g. Adaptive Size (Klinkenberg 2004) and FLORA2 (Widmer & Kubat 1996).

Instance weighting uses the ability of some learning algorithms such as Support Vector Machines to process weighted instances (Klinkenberg 2004). Instances can be weighted according to their age and competence with regard to the current concept. Klinkenberg (2004) shows that instance weighting techniques are worse at handling concept drift than analogous instance selection techniques, which is probably due to overfitting the data.

An ensemble learner combines the results of a number of classifiers, where each base (component) classifier is constructed on a subset of the available training instances. The research issues involved in using ensembles for handling concept drift involve first determining how to partition the instances into subsets with which to train the base classifiers. Then a mechanism for aggregating the results of the base classifiers must be determined. Finally, a mechanism for updating the ensemble to handle new instances and ‘forget’ older past instances must be established.

Building on the analysis presented in Kuncheva (2004) we propose that the techniques for using ensembles to handle concept drift fall into two groups:

- dynamic combiners where the base classifiers are trained in advance and the concept drift is tracked by changing the combination rule,
- incremental approaches that use fresh data to update the ensemble and incorporate an ensemble pruning or ‘forgetting’ mechanism to remove old or redundant data.

These approaches will be discussed below. It is worth noting that the two approaches are not mutually exclusive and combinations of both are possible.

Dynamic Combiners

The main techniques used for the dynamic combiners are variants on the Weighted Majority algorithm (Littlestone & Warmuth 1994) where the weights on the base classifiers are

altered based on how the base classifier performs as compared with the overall ensemble result. The issue with dynamic combiners is that the base classifiers are not re-trained with new instances so this approach is not appropriate for concept drift in spam as *new* types of spam are appearing and it is necessary to create new ensemble members.

Incremental Ensembles

The decision on how to partition the data into subsets with which to train the base classifiers is sometimes termed ‘data selection’. This decision will also determine how fresh instances are added into the ensemble. Kuncheva (2004) categorises three data selection approaches. The first reuses data points as is done in Bagging (random sampling with replacement) (Breiman 1996). The second approach to data selection is a filtering approach as in Boosting (Freund & Schapire 1999) or that used by Breiman (1999). The final data selection approach and the most common approach is one which uses blocks or chunks of data. These blocks normally group the data sequentially and could be of fixed size (Street & Kim 2001; Wang *et al.* 2003) or of variable size (Kolter & Maloof 2003; Stanley 2003).

Any incremental ensemble approach requires a *forgetting* or pruning strategy to identify which base classifiers should be dropped from the ensemble as new members are added. The simplest pruning strategy is to drop the oldest classifier once a new member has been added. More complex strategies are based on the actual performance of the base classifiers. Wang *et al.* (2003) keeps the top K base classifiers with the highest accuracy on the current training data block while Street and Kim (2001) favour the base classifiers that correctly classify instances (of the current block) on which the ensemble is ‘nearly undecided’. The worst performing classifier is replaced by the new member classifier. Stanley (2003) and Kolter and Maloof (2003) record the performance of each member against all seen instances and periodically remove those classifiers whose performance falls below a particular threshold.

In summary, it appears that the fixed framework of the ‘dynamic combiner’ approach is not appropriate for spam because as new types of spam emerge there is a need to create new ensemble members. Dynamic Weighted Majority (Kolter & Maloof 2003) attempts to resolve this problem by using the Weighted Majority algorithm, combining it with an update policy to create and delete base classifiers in response to changes in performance. On the other hand, the idea of using recent data to generate new ensemble members is clearly appropriate and the techniques we evaluate are variants on this idea. We also investigate the use of an ensemble pruning strategy based on dropping members that demonstrate poor performance on recent data.

Handling Concept Drift in Spam

Our previous work on handling concept drift in spam filtering presented an instance-selection approach that uses a case-based classifier (Delany *et al.* 2005). The case-based approach to filtering, known as Email Classification Using Examples (ECUE), involves setting up a case-base of training data selected from a user’s spam and legitimate email.

Details of the feature extraction and selection, case representation and case retrieval methods used are described in (Delany *et al.* 2005). The case-base maintenance procedure applied to allow ECUE to handle concept drift has two components; an initial case-base editing stage and a case-base update protocol.

The case-base that is built from the initial training data is edited using an editing technique called Competence Based Editing which, when applied to the spam domain, has shown to result in better generalisation accuracy than more traditional case editing techniques (Delany & Cunningham 2004). This editing technique is effective in this domain as it identifies and removes the cases in the case-base that cause other cases to be misclassified rather than the more common technique of removing cases that are actually misclassified.

The update procedure to update the case-base to allow it to handle concept drift in the emails is a two-phase procedure. First, any misclassified emails were added to the case-base daily. Then a feature reselection process is performed periodically to ensure that the case representation is updated to include features predictive of changes to the concept or data distribution (Delany *et al.* 2005). For the purposes of the evaluation in this paper, feature reselection was performed at the end of each month.

Since ensembles are a recognised technique for handling concept drift, it is important to evaluate this instance-selection approach against the ensemble alternatives for handling concept drift in spam filtering.

An Ensemble Approach

There are many approaches to generating and combining ensemble members $T = \{T_1, T_2, \dots, T_n\}$ as discussed previously. In order to construct an ensemble to compare against ECUE, which uses a nearest neighbour classifier, each individual ensemble member T_i used in this work is a nearest neighbour classifier built from a selection of the available training data. Each member uses k nearest neighbour (k NN) with $k = 3$, as is used in ECUE, and a distance weighted similarity measure (Mitchell 1997). Based on the accumulated similarity scores from all the k neighbours, each member T_i returns the result set $\{y_{ij} : 0 < y_{ij} < 1\}$ where y_{ij} is the score of member T_i for classification $c_j \in C$ (C is the set of all possible classifications, in our case here $C = \{spam, nonspam\}$). The y_{ij} 's are normalised such that $\sum_{j=1}^{|C|} y_{ij} = 1$.

The aggregation method used to determine the overall classification c_{AGG} from all ensemble members is the classification with the largest score after an accumulation of each classification result from each ensemble member; $c_{AGG} = \operatorname{argmax}_{j=1}^{|C|} (1/|T|) \sum_{i=1}^{|T|} y_{ij}$. This, in effect, is majority voting. The vote for each class, *spam* and *nonspam*, is normalised such that the sum of the votes adds to 1.

By comparing the vote for the *spam* class to a threshold t where $0 < t < 1$, this aggregation method has the advantage of allowing the ensemble to be biased away from FPs. Setting a threshold $t = 0.5$ is equivalent to the majority voting just described, but setting a threshold of, e.g. $t = 0.9$, would ensure that the normalised accumulated spam vote from all

member classifiers would have to be 0.9 or higher for the target email to be classified by the ensemble as spam. Setting a high value for t makes it more difficult for an email to be classified as spam thus reducing the FPs.

The main ensemble data selection approaches that we are presenting in this paper involve dividing the training data into blocks of fixed size organised by date and building an ensemble member using each block of training data. There are two main mechanisms used to partition the training data; a disjoint block selection mechanism which we call *Disjoint Date* and an overlapping mechanism which we call *Overlapping Date*. The Overlapping Date approach divides the training emails into overlapping sets where the percentage overlap between consecutive segments can be specified. In both approaches the number of ensemble members (i.e. the number of blocks or segments) is specified.

As ensemble pruning based on performance of the base classifiers on recent data has been successful, we also evaluate a mechanism which we call Context Sensitive Member Selection (CSMS) where context is defined by performance on the most recent block of training data, i.e. ensemble members with poor accuracy on recent training examples are discarded. We evaluate ensembles where members with error scores in the bottom half of the error range (using recent examples across all ensemble members) are dropped. We also considered a less severe regime where those in the bottom third of the range are dropped.

Evaluation

The objective of the evaluation is to compare the performance of ensemble approaches to handling concept drift against the ECUE approach which is an instance selection approach that operates on a single classifier. This section outlines the experimental setup and includes the results for both the static and dynamic stages of evaluation.

Experimental Setup

The static evaluation involved comparing the generalisation accuracy of the ensemble approaches and ECUE across 4 different datasets of 1000 emails each, using 10-fold cross validation. Each dataset consisted of 500 spam and 500 legitimate emails received by a single individual. The legitimate emails in each dataset include a variety of personal, business and mailing list emails.

The dynamic evaluation involved comparing the ensemble alternatives with ECUE using two further datasets of 10,000 emails each (described in (Delany *et al.* 2005)) that cover a period of one year. The first 1000 emails in each dataset were used as training data to build the initial classifier and the remaining emails, presented for classification in date order, were used for testing and updating the classifiers.

To summarise, the static evaluation employed 10-fold cross-validation while the dynamic evaluation was in effect an incremental validation. The static evaluation allows us to evaluate the effectiveness or discriminating power of an ensemble in the spam filtering domain while the dynamic evaluation allows us to evaluate how well an ensemble could handle the concept drift inherent in email.

Table 1: Results of static evaluation

Classifier	Average over 4 datasets			
	maj vote		with bias	
	%Err	%FPs	%Err	%FPs
Disjoint Sets (5 members)	6.7%	9.1%	10.6%	1.5%
Overlapping Sets (30% overlap; 5 members)	6.7%	8.9%	10.3%	1.7%
Bagging (20 members)	6.2%	8.9%	8.1%	2.7%
ECUE	4.4%	5.8%	5.5%	2.0%

Static Evaluation

Each dataset was evaluated (on the same cross validation folds) using three different data selection mechanisms; Bagging, Disjoint Sets and Overlapping Sets with a 30% overlap using between 5 and 20 ensemble members. We include Bagging as a baseline technique. As it was a static cross validation, the date order of the emails was not preserved in the ensemble member generation.

As expected, Bagging had a better generalisation accuracy with a larger number of ensemble members but Disjoint Sets and Overlapping Sets had a better generalisation accuracy with ensemble members of larger size, i.e. with a smaller number of ensemble members. The average results across the four datasets for each data selection method for the most successful ensemble size are displayed in Table 1. The results include figures for both the majority voting aggregation method (labeled *maj vote*) and aggregation involving a bias away from FPs with a threshold of 0.9 (labeled *with bias*).

Corresponding figures are also included for ECUE, with emails presented in random order, to allow comparisons between this and the ensemble approaches. There is limited scope to bias a k -NN classifier with $k = 3$. Requiring a unanimous vote (i.e. all returned neighbours to be of classification *spam*) for spam produces the max bias away from FPs. Higher values of k will allow a stronger bias; however $k = 3$ produces best overall accuracy. McNemar’s test (Dietterich 1998) was used to calculate the confidence levels between each ensemble method and ECUE to determine whether significant differences exist. These differences were significant at the 99.9% level in all cases except for the FPs figures for the bias results where the differences were not significant.

This evaluation shows that none of the selection of ensemble approaches improves on the accuracy of ECUE in the static setting. This is not surprising and is predicted by Breiman (1996) who points out that different training datasets will not produce diversity in ensembles of lazy learners (case-based classifiers) thus there will be no increase in accuracy.

One benefit arising from the ensemble approach is the potential to have greater control over the level of FPs with the ensemble than with the single classifier. Setting a threshold of $t = 0.9$ on the ensembles and using unanimous voting on ECUE produces better FP figures for the ensemble approaches than for ECUE, albeit at a considerable cost in FNs and therefore accuracy. However, it is clear from com-

parisons of the majority voting alternatives that the *discriminating* power of the ensembles is, if anything, worse than ECUE when there are equal misclassification costs for both classes (i.e. no bias).

Dynamic Evaluation

The dynamic evaluation used two large datasets as discussed previously. An update policy was used to regularly update the initial classifier. The update policy for the ECUE classifier is that described earlier. The ensemble update strategy is explained below. We evaluated both the Disjoint Date and the Overlapping Date ensemble data selection methods but not Bagging as it does not lend itself to an update policy to handle concept drift. Disjoint Date and Overlapping Date correspond to the Disjoint Sets and Overlapping Sets used in the static evaluation.

Ensemble Update Policy The update procedure for an ensemble involved adding new members to each ensemble up to a maximum of 10 members. At that stage the oldest existing member was dropped as a new member was added, maintaining the ensemble at a maximum of 10 members. New members had equal numbers of spam and legitimate email and were added once enough new emails had been processed to get the appropriate number for a new ensemble member. A common ensemble update technique is to use a measure of global error as a trigger to create a new ensemble member. This is not appropriate in this domain as it is unacceptable to wait until the spam filter performs badly before adding new training data. The filter should try to pro-actively anticipate the concept drift.

There is no standard class distribution for spam/legitimate emails, some individuals receive significantly more spam than legitimate email while for others it is the opposite. In addition, including a higher proportion of spam as training data will bias the classifier towards predicting spam which for this domain is unacceptable. For these reasons a balanced case-base was used for each ensemble member. This is supported by Weiss and Provost (2003) who conclude that a balanced distribution is a reasonable default training distribution when the natural distribution is not available.

As individuals normally do not receive equal numbers of spam and nonspam, the class with the larger number of emails during that time period was randomly sampled to select training data for the new ensemble member. Feature selection, based on Information Gain (Delany *et al.* 2005), was performed on each new ensemble member ensuring greater diversity between the members.

In addition to the Disjoint Date and the Overlapping Date data selection techniques, which use the simple ensemble update policy described above, we also evaluated the Disjoint Date data selection technique using the CSMS ensemble pruning strategy. This effectively incorporated a forgetting mechanism that was dependent on the performance of the base classifiers which is a typical ensemble pruning strategy (Street & Kim 2001; Wang *et al.* 2003). When a new member is to be added to the ensemble, those base classifiers that achieve a generalisation error of less than the average error across all base classifiers are dropped. The new

member is always added. One of the issues with the CSMS policy is that the number of base classifiers used in the ensemble tends towards two as new members are added to the ensemble. To evaluate whether leaving more base classifiers improves the performance we used a less severe policy that removed only those base classifiers that had a generalisation error that was less than two-thirds of the difference between the best and the worst error.

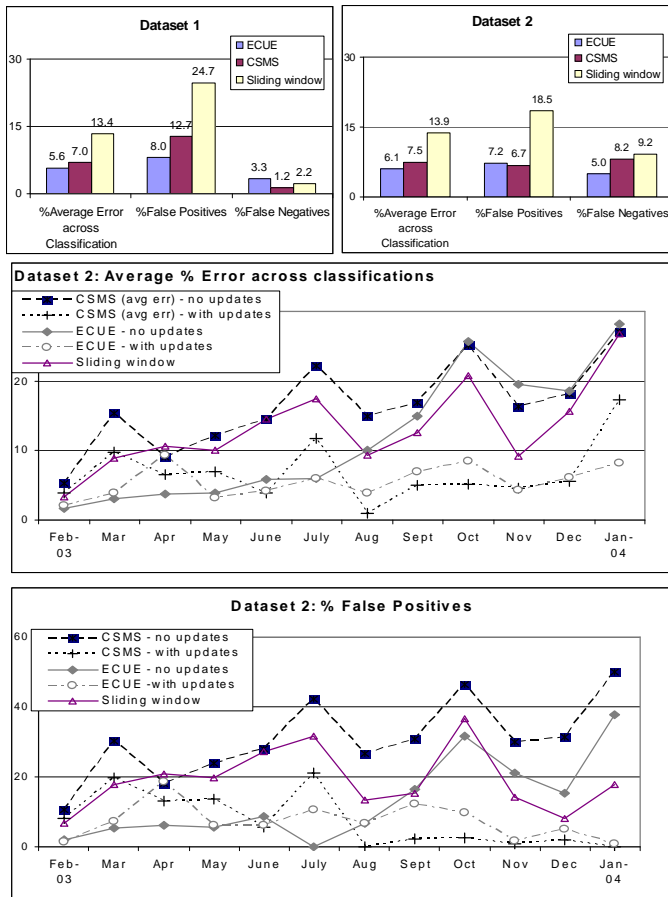


Figure 1: Effects of applying update policies to handle concept drift using ECUE, an ensemble of case-base classifiers and using a sliding window approach.

Results Figure 1 shows how concept drift is handled by both an ensemble of case-base classifiers (based on the CSMS policy) and ECUE. Emails were classified in date order against the training data and results were accumulated and reported at the end of each month. The overall results over the full datasets are reported. Given the significance of FPs in the spam filtering domain the evaluation metrics we are using here include the average error across the two classes $AvgErr = (\%FPs + \%FN)/2$ and the FP rate ($\%FP$). The average error is used as the numbers of spam and legitimate mail in the testing data are not equal (they were equal in the static evaluation). As the number of legitimate emails is considerably lower than spam email in these

Table 2: Results of dynamic evaluation

Classifier	Dataset 1				Dataset 2			
	maj vote		with bias		maj vote		with bias	
	Avg %Err	%FPs	Avg %Err	%FPs	Avg %Err	%FPs	Avg %Err	%FPs
Disjoint Date	10.6	16.9	8.8	1.4	8.7	14.3	18.5	0.8
Overlapping Date	10.6	16.4	7.6	2.2	9.0	10.5	20.7	0.9
CSMS (top 2/3)	9.4	16.2	9.3	1.5	8.3	6.9	21.0	0.6
CSMS (avg err)	7.0	12.7	10.0	1.2	7.5	6.7	19.2	0.6
ECUE	5.6	8.0	4.7	2.2	6.1	7.2	7.2	2.3
CSMS vs ECUE significance	No	No	Yes	Yes	No	No	Yes	Yes

datasets, the actual error figure would follow the False Negative (FN) rate and not give adequate emphasis to FPs.

Figure 1 includes a graph of the results for Dataset 2 which shows the results when no updates (labeled *no updates*) were applied to the initial training data and results with the appropriate update procedure in place (labeled *with updates*). The update policy for the single case-base classifier, ECUE, involved adding misclassified instances to the case-base regularly and performing periodic feature reselections while the ensemble update policy involved adding new ensemble members when adequate data has been received and pruning the ensemble using an assessment of the performance of the base classifiers on the most recent data. It is evident from these graphs that applying updates to the training data, for both types of classification process, helps to track the concept drift in the data. The figures suggest that ECUE appears to handle the concept drift marginally better than the ensemble.

Our earlier work on evaluating how ECUE handles concept drift (Delany *et al.* 2005) included an evaluation of applying a sliding window approach which is the most common technique for handling concept drift. This work concluded that ECUE performed better than the sliding window approach. Results of applying a sliding window approach are included in Figure 1 for comparison purposes. In order to compare with the ECUE update procedure, the window size was 1000, 500 spam and 500 non spam and the frequency of ‘the slide’ was monthly, i.e. at the start of each month a new training set was used.

Table 2 gives the overall results of the dynamic evaluation for the ensemble techniques and for ECUE including the results of a one tailed t-test (at the 95% level) which compared the monthly CSMS results with the ECUE results. Comparisons of the majority voting alternatives (i.e. no bias) show that the ECUE performs better than any of the ensemble techniques in terms of lower average error, albeit without a significant difference in all cases. ECUE also has a lower FP rate than the ensemble techniques except in the case of Dataset 2. The benefit evident from the static evaluation of the potential to have more control over the level of FPs is also evident here in the dynamic evaluation. Using the biasing policies previously described, the FP rates for the ensemble approaches are the same or better than ECUE in all

cases. However, even with these good FP rates the ensemble techniques have considerably higher average error rates than ECUE indicating a poor FN score.

The majority vote figures for the ensemble approaches show that the CSMS policy is the best performing of all the ensemble approaches. The more severe member deletion policy of removing all base classifiers less than the average error performs better than the moderate one of just removing those in the bottom third of the error range. This indicates that context sensitive ensemble pruning has merit. In effect, it is removing the base classifiers that are not effective in the ensemble. However, although approaching the non-bias results for ECUE, ECUE still has lower average error indicating that it has better discriminating power.

Conclusions

It is clear from the graphs presented in Figure 1 that spam filtering is a classification problem with significant concept drift. Our evaluations show that the case-base maintenance (instance selection) approach to handling concept drift is at least as effective as the ensemble alternatives we have evaluated and appears marginally better although the differences are not statistically significant. It is more straightforward and easier to handle concept drift with case-base maintenance rather than creating new classifiers as required by the ensemble approach. The most effective ensemble technique is one where the best ensemble members are selected based on an assessment of their performance on recent data. Some of the ensemble techniques return very strong results on False Positives; this comes at a significant cost in overall accuracy. We have pointed out that this reflects the greater potential there is to control the bias of the classifier away from FPs. In a sense, the strong performance of the case-editing technique is not surprising as it reflects the advantage of addressing concept drift at an instance level rather than at an ensemble member level.

Before giving up on the use of ensembles on this problem we propose to consider a more complex integration strategy. For instance a variant of dynamic integration as described by Tsymbal and Puuronen (2000) can be used. In addition we propose to evaluate weighted ensemble members such as those used by Kolter and Maloof (2003) and Stanley (2003) and build ensemble members that cover different parts of the problem space (most likely using a clustering algorithm) rather than depending on members that cover particular time periods.

References

- Breiman, L. 1996. Bagging predictors. *Machine Learning* 24(2):123–140.
- Breiman, L. 1999. Pasting small votes for classification in large databases and online. *Machine Learning* 36(1-2):85–103.
- Delany, S. J., and Cunningham, P. 2004. An analysis of case-based editing in a spam filtering system. In *7th Eur. Conf. on Case-Based Reasoning*, volume 3155 of *LNAI*, 128–141. Springer.
- Delany, S. J.; Cunningham, P.; Tsymbal, A.; and Coyle, L. 2005. A case-based technique for tracking concept drift in spam filtering. *Knowledge-Based Systems* 18(4–5):187–195.
- Dietterich, D. T. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computing* 10:1895–1923.
- Freund, Y., and Schapire, R. 1999. A short introduction to boosting. *Journal Japanese Society for Artificial Intelligence* 14(5):771–780.
- Kelly, M.; Hand, D.; and Adams, N. 1999. The impact of changing populations on classifier performance. In *KDD-99*, 367–371. ACM Press.
- Klinkenberg, R. 2004. Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis* 8(3).
- Kolter, J., and Maloof, M. 2003. Dynamic weighted majority: a new ensemble method for tracking concept drift. In *3rd IEEE Int. Conf. on Data Mining*, 123–130.
- Kubat, M., and Widmer, G. 1995. Adapting to drift in continuous domains. In *8th Eur Conf on Machine Learning*, volume 912 of *LNCS*, 307–310. Springer.
- Kuncheva, L. I. 2004. Classifier ensembles for changing environments. In *5th International Workshop on Multiple Classifier Systems (MCS 2004)*, 1–15. Springer.
- Littlestone, N., and Warmuth, M. 1994. The weighted majority algorithm. *Information and Computation* 108(2):212–261.
- Mitchell, T. 1997. *Machine Learning*. McGraw Hill.
- Salganicoff, M. 1997. Tolerating concept and sampling shift in lazy learning using prediction error context switching. *AI Review* 11(1-5):133–155.
- Stanley, K. 2003. Learning concept drift with a committee of decision trees. *Tech Rpt UT-AI-TR-03-302, Dept of Computer Science, University of Texas at Austin*.
- Street, W., and Kim, Y. 2001. A streaming ensemble algorithm (sea) for large-scale classification. In *KDD-01*, 377–382. ACM Press.
- Tsymbal, A., and Puuronen, S. 2000. Bagging and boosting with dynamic integration of classifiers. In *Proc of PKDD 2000*, 116–125. Springer.
- Vapnik, V. 1999. *The Nature of Statistical Learning Theory, 2nd. Ed.* Statistics for Engineering and Information Science. New York: Springer.
- Wang, H.; Fan, W.; Yu, P.; and Han, J. 2003. Mining concept-drifting datastreams using ensemble classifiers. In *KDD-03*, 226–235. ACM Press.
- Weiss, G., and Provost, F. 2003. Learning when training data are costly: The effect of class distribution on tree induction. *Journal of AI Research* 19:315–354.
- Widmer, G., and Kubat, M. 1996. Learning in the presence of concept drift and hidden contexts. *Machine Learning* 23(1):69–101.