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

Sarah Jane Delany[1], **Padraig Cunningham**[2], **Alexey Tsymbal**[2]
[1]Dublin Institute of Technology, Kevin St, Dublin 8
[2]Trinity College Dublin, Dublin 2

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 consider 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. We present an evaluation that shows that the case-base maintenance approach is more effective than a variety 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 classifer. However this comes at considerable cost in overall accuracy.

1 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 *et al.*, 1999]. There is 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. Subcategories 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 on concept drift shows that ensemble approaches are among the most effective [Kolter and Maloof, 2003; Kuncheva, 2004; Stanley, 2003; Street and 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 an approach that uses a case-base maintenance protocol with a single case-base classifier. 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 and periodically performs a feature reselection process to ensure that the new features are reflected in the case representation [Delany *et al.*, 2005a; 2005b].

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 case-base update approach using cross-validation. This evaluation showed that the ensembles that were considered did not improve on the classification accuracy of the 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 simulated online evaluation which compares the performance of the ensembles with that of a single case-base classifier over the period of a year 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 the best at tracking this concept drift. The evaluation also shows that an ensemble update 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. However, the evaluation suggests that a protocol for managing the training data of a single classifier will be more straightforward and at least as effective as an ensemble for tracking concept drift.

*This research was supported by funding from Enterprise Ireland under grant no. CFTD/03/219 and funding from Science Foundation Ireland under grant no. SFI-02IN.1I111

The body of this paper begins with a review of techniques for handling concept drift in Section 2. Then the approaches for handling concept drift in spam that are evaluated in this paper are described in Section 3. These alternative approaches are evaluated in Section 4 and the paper concludes with a summary and suggestions for future work in Section 5.

2 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 and Kubat, 1996], FRANN [Kubat and Widmer, 1994] and Time-Windowed Forgetting (TMF) [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 and Kubat, 1996].

Instance weighting uses the ability of some learning algorithms such as Support Vector Machines (SVMs) to process weighted instances [Klinkenberg, 2004]. Instances can be weighted according to their age and their competence with regard to the current concept. Klinkenberg [2004] shows that instance weighting techniques are worse at handling concept drift than analagous 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 an ensemble 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 a "forgetting" mechanism to remove old or redundant data from the ensemble

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

2.1 Dynamic Combiners

The main techniques used for the dynamic combiners are variants on the Weighted Majority algorithm [Littlestone and 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.

2.2 Incremental Ensembles

The decision on how to partition the data into subsets with which to train the base classifiers sometimes is termed 'data selection'. This decision will also determine how fresh instances are added into the ensemble. Kuncheva categorises three data handling 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 and 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 e.g. [Street and Kim, 2001; Wang *et al.*, 2003] or of variable size e.g. [Kolter and Maloof, 2003; Stanley, 2003].

Any incremental ensemble approach requires a *forgetting* mechanism to identify which base classifiers should be dropped from the ensemble as the new members are added. The simplest forgetting 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 chunk 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 classifer 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 who performance falls below a particular threshold.

2.3 Appropriate Techniques for Spam Filtering

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 and Maloof, 2003] attempts to resolve this problem by using the Weighted Majority algorithm but 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 test the idea of dropping ensemble members with poor performance on recent data and this proves effective.

3 Handling Concept Drift in Spam

Previous work on handling concept drift in spam filtering [Delany *et al.*, 2005a] presented an instance-selection approach that used a case-based classifier. This approach is

summarised below in Section 3.1. 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.

3.1 A Case-based Approach

The case-base approach to filtering, known as Email Classification Using Examples (ECUE), which is used in this paper 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.*, 2005b]. The case-base maintenance procedure used 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 and Cunningham, 2004]. This editing technique is effective in this domain as it attempts to identify and remove the cases in the case-base that caused other cases to be misclassified rather than the more common technique of removing cases that were actually misclassified.

The update procedure to update the case-base to allow it to handle any 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 the case representation is updated to reflect features predictive of changes to the concept or data distribution [Delany *et al.*, 2005a].

3.2 An Ensemble Approach

There are many approaches to generating and combining ensemble members $T = \{T_1, T_2, \ldots, T_n\}$ as outlined in Section 2. 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 (kNN) 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 for member T_i, for classification c_j ($c_j \in C$ is the set of all possible classifications). 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 a straighforward accumulation of each classification result from each ensemble member, $c_{AGG} = \text{argmax}_{i=1}^{|C|} 1/|T| \sum_{j=1}^{|T|} y_{ij}$ This, in effect, is (weighted) majority voting. The vote for each class, *spam* and *nonspam*, is normalised such that the sum of the votes add 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.

A more common approach is where member T_i returns the winning classification y_j rather than a numeric score for each classification as described above. The aggregation method in this situation is simply $C_{AGG} = \text{argmax}_i \sum_{j=1}^{|T|} 1(y_j, c_i)$ where $1(y_j, c_i)$ returns 1 if $y_j = c_i$. Our cross validation experiments indicated that the generalisation error of this aggregation method is higher than using the numeric result of each ensemble member in the aggregation.

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 chunks or segments) is specified.

We also evaluate a member selection 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 only those in the bottom third of the range are dropped.

4 Evaluation

The objective of the evaluation is to compare the performance of ensemble approaches to concept drift against the ECUE approach which is an instance selection approach that operates on a single classifier. To separate 'ensemble effects' from 'concept drift effects' two types of evaluation were performed, a cross-validation comparison of the ensemble alternatives against ECUE on static datasets and a dynamic evaluation using date-ordered datasets which was in effect a simulated online evaluation. This section outlines the experimental setup and includes the results for both the static and dynamic stages of evaluation.

4.1 Experimental Setup

The static evaluation involved comparing the generalisation accuracy of different ensemble approaches and ECUE across 4 different datasets of 1000 emails each. 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.

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%

The evaluation involved performing a 10-fold cross validation on each dataset.

The dynamic evaluation involved comparing the ensemble alternatives with ECUE using two further datasets of 10,000 emails each that covered a period of one year. These datasets are described in [Delany *et al.*, 2005a]. The first 1000 emails (consisting of 500 spam and 500 legitimate emails) in each dataset were used as training data to build the initial classifier and the remaining emails were used for testing. The test emails were presented for classification in date order. 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.

To summarise, the static evaluation employed 10-fold cross-validation while the dynamic evaluation was effectively an evaluation on unseen data. 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.

4.2 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 cross validation, the date order of the emails was not preserved in the ensemble member generation, so Disjoint Sets and Overlapping Sets correspond to Disjoint Date and Overlapping Date in the dynamic evaluation coming up in the next section.

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*) which is described in Section 3.2.

Balanced and biased figures for ECUE are also included 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 for spam produces the maximum bias. 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. The differences between each ensemble technique and ECUE were signifiant at the 99.9% level in all cases except for the FPs figures for the bias results where the differences were not significant in all cases.

This evaluation shows that none of the selection of ensemble approaches improves on the accuracy of ECUE. 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.

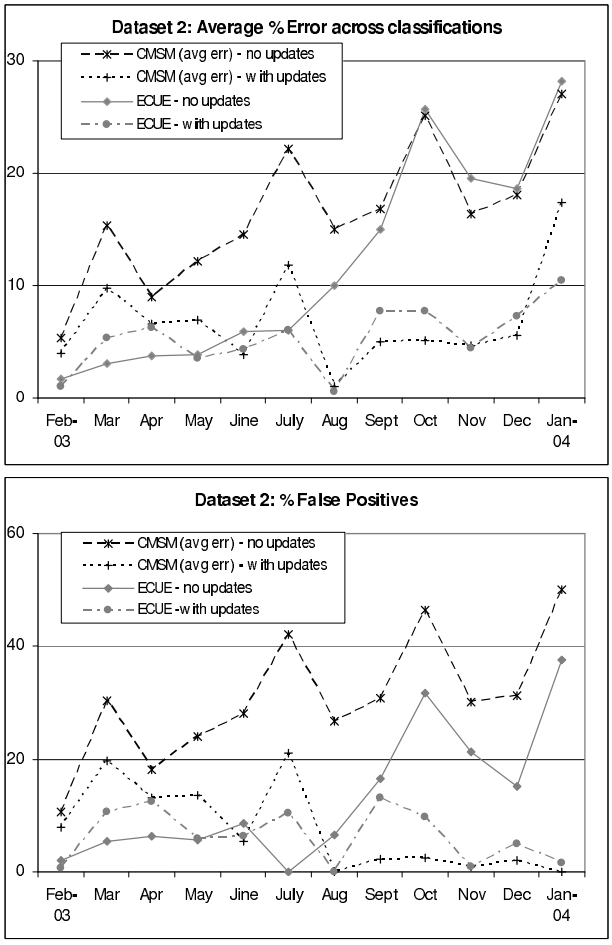

Figure 1: Effects of update policies on ECUE and an ensemble of case-base classifiers.

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 comparisons of the majority voting alternatives (i.e. no bias) that the

discriminating power of the ensembles is, if anything, worse than ECUE.

4.3 Dynamic Evaluation

As stated above, the dynamic evaluation used two large datasets; each was derived from email received by individuals over the period of approximately 1 year. The initial 1000 emails (500 spam and 500 legimate) from each of these datasets were used to build the initial ensemble and ECUE classifiers. The remaining data, including varying numbers of spam and legimate mail, was then used for testing. An update policy was used to regularly update the initial classifier. The update policy for the ECUE classifier is that described in Section 3.1. The ensemble update procedure is explained below.

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 the appropriate number of new emails had been processed. 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.*, 2005b], 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 mechanisms, we also evaluated the Disjoint Date data selection technique using the CSMS member selection policy described in Section 3.2. This effectively incorporated a forgetting mechanism that was dependent on the performance of the base classifiers. When a new member is to be added to the ensemble, those base classifiers that achieve a generalisation error of less that 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 2 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.

Results

Figure 1 shows, for one of the datasets, how concept drift is handled by both an ensemble of case-based classifiers 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 graph 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*). It is evident from the graph that applying updates to the training data, for both types of classification process, helps to track the concept drift in the data. It is also clear from comparisons of the ensemble and ECUE graphs that ECUE appears to handle the concept drift better than the ensemble.

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
Overlap Date (30% overlap)	10.6	16.4	7.6	2.2	9.0	10.5	20.7	0.9
CSMS (avg err)	7.0	12.7	10.0	1.2	7.5	6.7	19.2	0.6
CSMS (top $2/3$)	9.4	16.2	9.3	1.5	8.3	6.9	21.0	0.6
ECUE	6.4	10.0	4.7	2.2	6.0	6.6	7.2	2.5

Table 2 gives the results of the dynamic evaluation for the ensemble techniques and for ECUE. 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. As the number of legitimate emails is considerable lower than spam email in these datasets, the actual error figure would follow the False Negative (FN) rate and not give adequate emphasis to FPs.

Comparisons of the majority voting alternatives (i.e. no bias) show that the ECUE performs better than any of the ensemble techniques both in terms of lower average error and lower FP rate. 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 biasing policies described in Section 4.2, 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 selection 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.

5 Conclusions

It is clear from the graphs presented in Figure 1 that spam filtering is a classification problem with significant concept drift. The evaluation presented in Section 4.3 shows that the case-editing (instance selection) approach to handling concept drift is more effective than the ensemble alternatives we have evaluated. The discriminating power of the single

classifier solution is better than that of the ensemble techniques. 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 ensemble.

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.

5.1 Future Work

We are currently working on mechanisms for attaching confidence measures to predictions so that the set of items classified as spam can be partitioned into 'definitely spam' and possibly spam'. This will make the task of watching for False Positives much easier.

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.

References

[Breiman, 1996] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[Breiman, 1999] L. Breiman. Pasting small votes for classification in large databases and online. *Machine Learning*, 36(1-2):85–103, 1999.

[Delany and Cunningham, 2004] S. J. Delany and P. Cunningham. An analysis of case-based editing in a spam filtering system. In P.Funk and P.Gonzalez Calero, editors, *7th European Conference on Case-Based Reasoning, LNAI 3155*, pages 128–141. Springer, 2004.

[Delany *et al.*, 2005a] S. J. Delany, P. Cunningham, A. Tsymbal, and L. Coyle. A case-based technique for tracking concept drift in spam filtering. *Knowledge-Based Systems (to appear)*, 2005.

[Delany *et al.*, 2005b] S.J. Delany, P. Cunningham, and L. Coyle. An assessment of case-based reasoning for spam filtering. *Artificial Intelligence Review (to appear)*, 2005.

[Dietterich, 1998] D. T. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computing*, 10:1895–1923, 1998.

[Freund and Schapire, 1999] Y. Freund and R. Schapire. A short introduction to boosting. *J. Japan. Soc. for Artificial Intelligence 14(5)*, 14(5):771–780, 1999.

[Kelly *et al.*, 1999] M.G. Kelly, D.J. Hand, and N.M. Adams. The impact of changing populations on classifier performance. In *5th International Conference on Knowledge Discovery and Data Mining*, pages 367–371. ACM Press, 1999.

[Klinkenberg, 2004] R. Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, 8(3), 2004.

[Kolter and Maloof, 2003] J.Z. Kolter and M.A. Maloof. Dynamic weighted majority: a new ensemble method for tracking concept drift. In *3rd IEEE International Conference on Data Mining*, pages 123–130. IEEE CS Press, 2003.

[Kubat and Widmer, 1994] M. Kubat and G. Widmer. Adapting to drift in continuous domains. *Technical Report OFAI-TR-94-27, Austrian Research Institute for Artificial Intelligence, Vienna*, 1994.

[Kuncheva, 2004] L. I. Kuncheva. Classifier ensembles for changing environments. In F. Roli, J. Kittler, and T. Windeatt, editors, *Multiple Classifier Systems, 5th International Workshop, MCS 2004*, pages 1–15. Springer, 2004.

[Littlestone and Warmuth, 1994] N. Littlestone and M.K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[Mitchell, 1997] T. Mitchell. *Machine Learning*. McGraw Hill, New York, 1997.

[Salganicoff, 1997] M. Salganicoff. Tolerating concept and sampling shift in lazy learning using prediction error context switching. *AI Review*, 11(1-5):133–155, 1997.

[Stanley, 2003] K.O. Stanley. Learning concept drift with a committee of decision trees. *Technical Report UT-AI-TR-03-302, Department of Computer Science, University of Texas at Austin*, 2003.

[Street and Kim, 2001] W. Street and Y. Kim. A streaming ensemble algorithm (sea) for large-scale classification. In *7th International Conference on Knowledge Discovery and Data Mining*, pages 377–382. ACM Press, 2001.

[Tsymbal and Puuronen, 2000] A. Tsymbal and S. Puuronen. Bagging and boosting with dynamic integration of classifiers. In D.A. Zighed, H.J. Komorowski, and J.M. Zytkow, editors, *4th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 116–125. Springer, 2000.

[Vapnik, 1999] V. Vapnik. *The Nature of Statistical Learning Theory, 2nd. Ed.* Statistics for Engineering and Information Science. Springer, New York, 1999.

[Wang *et al.*, 2003] H. Wang, W. Fan, P.S. Yu, and J. Han. Mining concept-drifting datastreams using ensemble classifiers. In *9th International Conference on Knowledge Discovery and Data Mining*, pages 226–235. ACM Press, 2003.

[Widmer and Kubat, 1996] G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(1):69–101, 1996.