Learning Drifting Concepts with Partial User Feedback

Ralf Klinkenberg*

* Artificial Intelligence Unit, Computer Science Department, University of Dortmund,
Baroper Str. 301, D-44221 Dortmund, Germany
E-Mail: klinkenberg@ls8.cs.uni-dortmund.de
WWW: http://www-ai.cs.uni-dortmund.de/

Abstract. The task of information filtering is to classify texts from a stream of documents into relevant and irrelevant, respectively, with respect to a particular category or user interest, which may change over time. A filtering system should be able to adapt to such concept changes and to cope the problem of users giving only partial feedback. This paper explores methods to recognize concept changes and to maintain windows on the training data, whose size is either fixed or automatically adapted to the current extent of concept change. Experiments with two simulated concept drift scenarios based on real-world text data and four learning methods are performed to evaluate three indicators for concept changes and to compare approaches with fixed and adjustable window sizes, respectively, to each other and to learning on all previously seen examples. Additional experiments test the adaptive window size approach with four simulated user behaviours with partial feedback in the two aforementioned scenarios. Even using only a simple window on the data already improves the performance of the classifiers significantly as compared to learning on all examples. For most of the classifiers, the window adjustments lead to a further increase in performance compared to windows of fixed size. The chosen indicators allow to reliably recognize concept changes, even if only partial user feedback is available.

Keywords. Machine Learning, Adaptive Information Filtering, Text Classification, Concept Drift, Partial User Feedback

1 Introduction

With the amount of online information and communication growing rapidly, there is an increasing need for automatic information filtering. Information filtering techniques are used, for example, to build personalized news filters, which learn about the news-reading preferences of a user ([13], [21]), or to filter e-mail [6]. The concept underlying the classification of the texts into relevant and irrelevant may change. Machine learning techniques ease the adaptation to (changing) user interests.

This paper focuses on the aspect of changing concepts in information filtering. After reviewing the standard feature vector representation of text and giving some references to other work on adaptation to changing concepts, this paper describes indicators for recognizing concept changes and uses some of them as a basis for a window adjustment heuristic that adapts the size of a time window on the training data to the current extent of concept change [9], [10]. The indicators and data management approaches with windows of fixed and adaptive size are evaluated in two simulated concept drift scenarios on real-world text data. As most real information system users do not provide feedback for all incoming documents, the applicability of the adaptive window size approach is investigated in additional experiments with the partial feedback of four simulated users in both concept drift scenarios.

2 Text Representation

In Information Retrieval, words are the most common representation units for text documents and it is usually assumed, that their ordering in a document is of minor importance for many tasks. This leads to an attribute-value

representation of text, where each distinct word w_i corresponds to a feature with the number of times it occurs in the document d as its value (*term frequency*, $TF(w_i, d)$). The length of the feature vector is reduced by considering only words as features that occur at least 3 times in the training data and are not in a given list of stop words (like "the", "a", "and", etc.).

For some of the learning methods used in the experiments described in this paper, a subset of the features is selected using the *information gain* criterion [18], to improve the performance of the learner and/or speed up the learning process. The remaining components w_i of the document feature vector are then weighted by multiplying them with their *inverse document frequency (IDF)*. Given the *document frequency* $DF(w_i)$, i. e. the number of documents word w_i occurs in, and the total number of documents $|D|$, the inverse document frequency of word w_i is computed as $IDF(w_i) = \log \frac{|D|}{DF(w_i)}$. Afterwards each document feature vector is normalized to unit length to abstract from different document lengths.

2.1 Performance Measures

In the experiments described in this paper, the performance of a classifier is measured by the three metrics accuracy, recall, and precision. *Accuracy* is the probability, that a random document is classified correctly. It is estimated as the number of correct classifications divided by the total number of classifications. *Recall* is the probability, that the classifier recognizes a relevant document as relevant, and is computed as the number of relevant documents classified as relevant divided by the total number of relevant documents. *Precision* is the probability, that a document classified as relevant actually is relevant. It is estimated by the number of relevant documents classified as relevant divided by the total number of documents classified as relevant. The metrics can be computed from a contingency table:

	Relevant	Irrelevant
Classified as relevant	a	b
Classified as irrelevant	c	d

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} \tag{1}$$

$$\text{Recall} = \frac{a}{a+c} \tag{2}$$

$$\text{Precision} = \frac{a}{a+b} \tag{3}$$

3 Adapting to Changing Concepts

In machine learning, changing concepts are often handled by time windows of fixed or adaptive size on the training data (see for example [22], [14]) or by weighting data or parts of the hypothesis according to their age and/or utility for the classification task ([11], [19]). The latter approach of weighting examples has already been used in information filtering by the incremental relevance feedback approach [1] and by [2]. In this paper, the earlier approach maintaining a window of adaptive size and explicitly recognizing concept changes is explored in the context of information filtering. More detailed descriptions of the methods described above and further approaches can be found in [9].

For windows of fixed size, the choice of a *"good" window size* is a compromise between fast adaptability (small window) and good and stable learning results in phases without concept change (large window). The basic idea of the *adaptive window management* is to adjust the window size to the current extent of concept drift. In case of a suspected concept drift or shift, the window size is decreased by dropping the oldest, no longer representative training instances. In phases with a stable concept, the window size is increased to provide a large training set as basis for good generalizations and stable learning results. Obviously, reliable indicators for recognizing concept changes play a central role in such an adaptive window management.

3.1 Indicators for Concept Drifts

Different types of indicators can be monitored to detect concept changes:

- *Performance measures* (e. g. the accuracy of the current classifier): independent of the hypothesis language, generally applicable, but requiring user feedback.

- *Properties of the classification model* (e. g. the complexity of the current rules): dependent on a particular hypothesis language, applicable only to some classifiers.
- *Properties of the data* (e. g. class distribution, attribute value distribution, current top attributes according to a feature ranking criterion, or current characteristic of relevant documents like cluster memberships): independent of the hypothesis language, generally applicable.

The indicators of the window adjustment heuristic of the FLORA algorithms [22], for example, are accuracy and coverage of the current concept description, where coverage is the number of positive instances covered by the current hypothesis divided by the number of literals in this hypothesis. Obviously the coverage can only be computed for rule-based classifiers. The SIFTER information filtering system [12] determines clusters of similar documents and monitors the probability to be relevant for documents of each cluster seperately. Changes in the user interest are recognized by changes of these probabilities. The performance of the classifier is not monitored. This approach is independent of the learning method underlying the system.

The window adjustment approach for text classification problems proposed in this paper (and previously in [9], [10]) only uses performance measures as indicators, because they can be applied across different learning methods and are expected to be the most reliable indicators. The computation of performance measures like accuracy requires user feedback about the true class of filtered documents. In some applications only partial user feedback is available to the filtering system. While the experiments described in [9] and [10] assumed complete feedback about all filtered documents, this paper reports additional more realistic experiments with partial feedback. In most information filtering tasks, the irrelevant documents significantly outnumber the relevant documents and a default rule predicting all new documents to be irrelevant achieves a high accuracy, because accuracy does not distinguish between different types of misclassifications. As equation (1) shows, the accuracy does not only depend on a, but also on d, the number of irrelevant documents classified correctly. If d is assumed to be a very large, constant number, the accuracy does not reflect concept changes as much as recall and precision (equations (2) and (3)). Hence accuracy alone is only of limited use as performance metric and indicator for text classifiers. Therefore the metrics recall and precision are used as indicators in addition to accuracy, because they assess the performance on the smaller, usually more important class of relevant documents.

3.2 Adaptive Window Adjustment

The texts are presented to the filtering system in batches. Each batch is a sequence of several texts from the stream of documents to be filtered. In order to recognize concept changes, the values of the three indicators accuracy, recall, and precision are monitored over time and the average value and the standard sample error are computed for each of these indicators based on the last M batches at each time step, where M is a user-defined constant. Each indicator value is compared to a confidence interval of α times the standard error around the average value of this indicator. The confidence niveau α is a user-defined constant ($\alpha > 0$). If the indicator value is smaller than the lower end point of this interval, a concept change is suspected. In this case, a further test determines, whether the change is abrupt and radical (*concept shift*) or rather gradual and slow (*concept drift*). If the current indicator value is smaller than its predecessor times a user-defined constant β ($0 < \beta < 1$), a concept shift is suspected, otherwise a concept drift.

In case of a concept shift, the window is reduced to its minimal size, the size of one batch ($|B|$), in order to drop no longer representative old examples as fast as possible. If only a concept drift has been recognized, the window is reduced less radically by a user-defined reduction rate γ ($0 < \gamma < 1$). Thereby some of the old, still at least partially representative data for the current concept is kept. This establishes a compromise between fast adaptivity via a reduction of the window size and stable learning results as a result of a large training data set. If neither a concept shift nor a drift is suspected, all seen examples are stored, in order to provide a training set of maximal size, because in case of a stable concept, text classifiers usually perform the better, the more training examples they have.

While in real applications an upper bound for the size of the adaptive window seems reasonable, no such bound was used for the experiments described in this paper. Figure 1 describes the window adjustment heuristic. For the first M_0 initial batches, the window size is not adapted, but left at its initial value of $|W_0|$ to establish the average indicator values and their standard errors. $|W_t|$ denotes the current window size and $|W_{t+1}|$ the new window size. $|B|$ is the number of documents in a batch. Acc_t is the current accuracy value, Acc_{t-1} is the previous accuracy value, $Avg_M(Acc)$ is the average accuracy of the last M batches, and $StdErr_M(Acc)$ is the standard error of the accuracy on the last M batches. Rec_t, Rec_{t-1}, $Avg_M(Rec)$, and $StdErr_M(Rec)$ denote the corresponding recall values, and $Prec_t$, $Prec_{t-1}$, $Avg_M(Prec)$, and $StdErr_M(Prec)$ the corresponding precision values.

Procedure DetermineNewWindowSize $(|W_t|, M, \alpha, \beta, \gamma)$

if $((Acc_t < Avg_M(Acc) - \alpha \cdot StdErr_M(Acc))$ **and** $(Acc_t < \beta \cdot Acc_{t-1}))$ **or**
$((Rec_t < Avg_M(Rec) - \alpha \cdot StdErr_M(Rec))$ **and** $(Rec_t < \beta \cdot Rec_{t-1}))$ **or**
$((Prec_t < Avg_M(Prec) - \alpha \cdot StdErr_M(Prec))$ **and** $(Prec_t < \beta \cdot Prec_{t-1}))$

then $|W_{t+1}| := |B|$; /* concept shift suspected: reduce window size to one batch */

else if $(Acc_t < Avg_M(Acc) - \alpha \cdot StdErr_M(Acc))$ **or**
$(Rec_t < Avg_M(Rec) - \alpha \cdot StdErr_M(Rec))$ **or**
$(Rec_t < Avg_M(Prec) - \alpha \cdot StdErr_M(Prec))$

then $|W_{t+1}| := \max(|B|, \ |W_t| - \gamma \cdot |W_t|)$; /* concept drift suspected: reduce window size by $\gamma \cdot 100\%$ */

else $|W_{t+1}| := |W_t| + |B|$; /* stable concept suspected: grow window by one batch */

return $|W_{t+1}|$;

Figure 1 Window adjustment heuristic for text categorization problems.

Category	Name of the Category	Number of Documents
1	Antitrust Cases Pending	400
3	Joint Ventures	842
4	Debt Rescheduling	355
5	Dumping Charges	483
6	Third World Debt Relief	528
	Total	2608

Table 1 Categories of the TREC data set used in the experiments.

Cate-gory	Probability of being relevant for a document of the specified category at the specified time step (batch)																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 2 Relevance of the categories in concept change scenario A (abrupt concept shift in batch 10).

Cate-gory	Probability of being relevant for a document of the specified category at the specified time step (batch)																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.6	0.8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 3 Relevance of the categories in concept change scenario B (slow concept drift from batch 8 to batch 12).

4 Experiments

4.1 Data Set, Simulated Concept Drift Scenarios, and Simulated User Feedback Behaviors

The experiments use a subset of the data set of the *Text REtrieval Conference (TREC)* consisting of English business news texts. Each text is assigned to one or several categories. Table 1 shows the names and sizes of the categories 1, 3, 4, 5, and 6 used here. For the experiments, two concept change scenarios are simulated. The texts are randomly split into 20 batches of equal size containing 130 documents each[1]. The texts of each category are distributed as equally as possible to the 20 batches.In the first scenario (*scenario A*), first documents of category 1 (Antitrust Cases Pending) are considered relevant for the user interest and all other documents irrelevant. This changes abruptely (concept shift) in batch 10, where documents of category 3 (Joint Ventures) are relevant and all others irrelevant. Table 2 specifies the probability of being relevant for documents of category 1 and 3 for each time step (batch). In the second scenario (*scenario B*), again first documents of category 1 (Antitrust Cases Pending) are considered relevant for the user interest and all other documents irrelevant. This changes slowly (concept drift) from batch 8 to batch 12, where documents of category 3 (Joint Ventures) are relevant and all others irrelevant. Table 3 specifies the probability of being relevant for documents of category 1 and 3 for each time step (batch). Classes 4, 5, and 6 are never relevant.

In a first set of experiments, the adpative window size approach is compared to alternative approaches in these two scenarios assuming complete feedback (sections 4.2, 4.3 and 4.4). A second set of experiments tests the adaptive window size approach with several different simulated user feedback behaviors on these scenarios (table 4 and section 4.5).

[1] Hence, in each trial, out of the 2608 documents 8 randomly selected texts are not considered.

User type	User 0	User 1	User 2	User 3	User 4
Probability of feedback for documents predicted to be positive	1.0	1.0	0.5	0.5	0.1
Probability of feedback for documents predicted to be negative	1.0	0.1	0.1	0.0	0.0

Table 4 Feedback behavior of the simulated users with complete (user 0) and partial feedback (users 1 to 4).

		Full Memory User 0	No Memory User 0	Fixed Size User 0	Adaptive Size User 0	Adaptive Size User 1	Adaptive Size User 2	Adaptive Size User 3	Adaptive Size User 4
Naive Bayes	Accuracy	82.26%	93.77%	92.40%	94.20%	88.49%	84.94%	84.07%	75.36%
	Recall	68.41%	87.04%	85.57%	88.80%	77.28%	70.49%	72.99%	42.47%
	Precision	68.10%	87.95%	85.22%	88.06%	77.12%	71.53%	70.29%	46.55%
SVM	Accuracy	79.65%	92.60%	91.77%	94.40%	87.37%	83.66%	84.92%	76.44%
	Recall	51.98%	74.55%	77.42%	84.30%	69.76%	60.50%	70.29%	36.05%
	Precision	64.77%	90.81%	86.42%	90.72%	78.89%	72.52%	72.84%	46.62%
C4.5	Accuracy	78.17%	90.90%	90.03%	92.93%	86.13%	81.53%	84.15%	75.32%
	Recall	45.75%	77.00%	74.37%	82.43%	68.71%	57.43%	73.16%	37.72%
	Precision	50.08%	82.38%	82.00%	86.66%	77.01%	67.80%	70.64%	45.19%
CN2	Accuracy	78.92%	90.00%	89.74%	92.48%	84.68%	81.28%	81.72%	75.81%
	Recall	42.17%	66.68%	67.47%	75.38%	62.60%	47.31%	65.29%	27.86%
	Precision	62.53%	84.03%	86.62%	90.78%	75.75%	69.53%	66.80%	44.62%

Table 5 Accuracy, recall and precision of all learning methods combined with all data management approaches for scenario A averaged over 10 trials with 20 batches each.

4.2 Experimental Setup

The experiments are performed according to the batch learning scenario, i. e. the learning methods learn a new classification model whenever they receive a new batch of training documents. In the first set of experiments, each of the following *data management approaches* is tested in combination with each of the learning methods listed further below:

- *"Full Memory"*: The learner generates its classification model from all previously seen examples, i.e. it cannot "forget" old examples.
- *"No Memory"*: The learner always induces its hypothesis only from the least recently seen batch. This corresponds to using a window of the fixed size of one batch.
- Window of *"Fixed Size"*: A window of the fixed size of three batches is used.
- Window of *"Adaptive Size"*: The window adjustment heuristic (figure 1) is used to adapt the window size to the current concept drift situation.

For the adaptive window management approach, the initial window size is set to three batches ($|W_0| := 3 \cdot |B|$), the number of initial batches to five ($M_0 := 5$), and the number of batches for the averaging process to 10 ($M := 10$). The width of the confidence interval is set to $\alpha := 5.0$, the factor $\beta := 0.5$, and the window reduction rate $\gamma := 0.5$. These values are arbitrarily set and not result of an optimization, but as emperically shown in [10], the window adjustment heuristic is fairly robust to changes of the parameters.

The parameters of the *learning methods* listed below were found to perform well in a prelimenary experiment for a different classification task on the TREC data set, but are not optimized for the concept drift scenarios considered here: a *Naive Bayes Classifier* [16], a *Support Vector Machine (SVM)* ([20], [7]) with polynomial kernel and polynom degree one (= linear kernel), the symbolic rule learner *CN2* ([5], [4]) with the default paramters for unordered rules, and the symbolic decision tree and rule learning system *C4.5* [18] with the default parameters to induce a decision tree, transform it to an ordered rule set, and post-prune the resulting rules. For C4.5 and CN2 the 1000 best attributes according to the information gain criterion [18] are selected. The other methods used all attributes. The results reported in the following sections are averaged over 10 trials for each combination of learning method, data management approach, and concept drift scenario.

4.3 Comparing Data Management Approaches with Complete Feedback on Scenario A (Concept Shift)

Columns 2 to 5 of table 5 show accuracy, recall, and precision of all combinations of learning methods and data management approaches averaged over 10 trials with 20 batches each according to scenario A (table 2) with complete

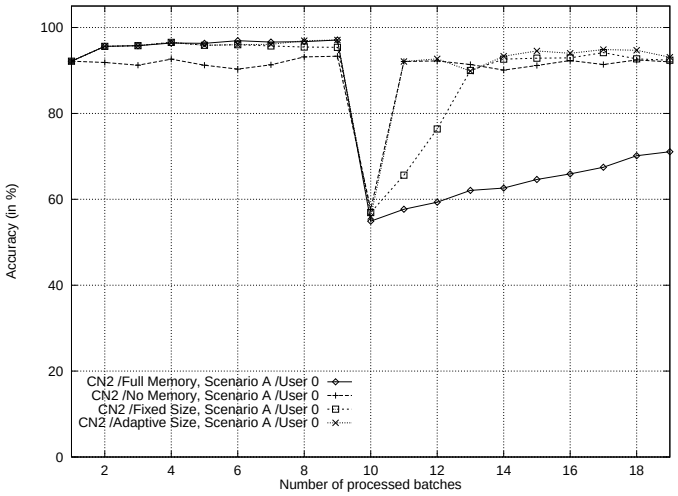

Figure 2 Accuracy of CN2 with the different data management approaches for scenario A averaged over 10 trials with 20 batches each.

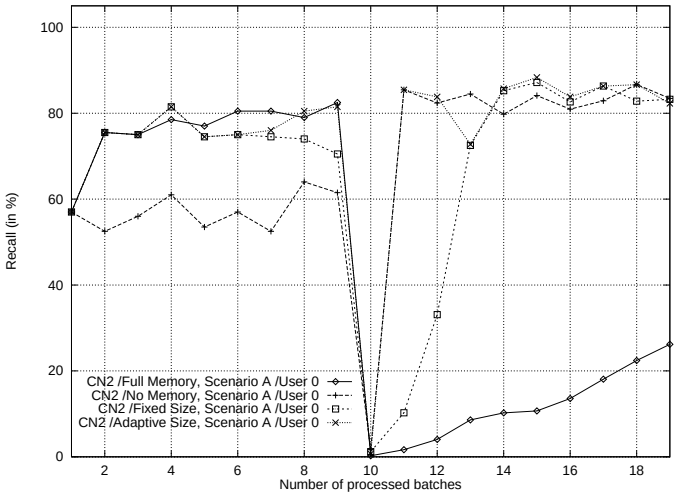

Figure 3 Recall of CN2 with the different data management approaches for scenario A averaged over 10 trials with 20 batches each.

user feedback (user 0). The columns 6 to 9 contain results from additional experiments described later in section 4.5. The results in table 5 demonstrate that even the simple approaches No Memory and Fixed Size with time windows of fixed size significantly outperform the approach learning on all previously seen examples (Full Memory). A further performance improvement is achieved by using the Adaptive Size approach instead of the best investigated approach with fixed window size.

Figures 2 to 4 show the values of the three indicators over time for the learning method CN2 in combination with all data management approaches and thereby allow a more detailed analysis of the results than table 5. The figures 2 to 4 with the accuracy, recall, and precision values of CN2 show two things. First, in this scenario all three indicators can be used to easily detect the concept shift, because their values decrease very significantly in the batch the shift occurs in (batch 10). Recall and precision indicate the shift even more clearly than accuracy.

Second, in this scenario the data management approaches demonstrate their typical behaviour in relation to each other. Before the shift, the Full Memory approach has the advantage of the largest training set and hence shows the most stable performance and outperforms the other three approaches, but recovers only very slowly from its break-down after the concept shift. The Fixed Size approach shows a relatively good performance in phases with stable target concept, but needs several batches to recover after the concept shift. The No Memory approach offers the maximum flexibility and recovers from the shift after only one batch, but in phases with a stable concept, this approach is less stable and performs worse than the other approaches. In this scenario and in combination with CN2, the Adaptive Size approach obviously manages to show a high and stable performance in stable concept phases *and* to adapt very fast to the concept shift. Hence Adaptive Size here is able to combine the advantages of different window sizes.

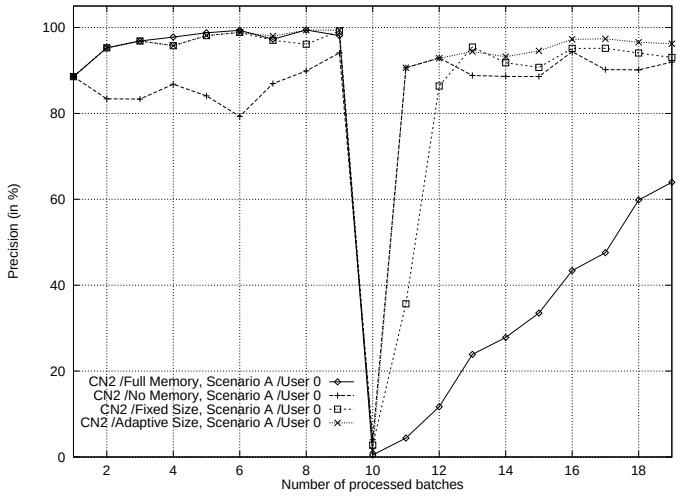

Figure 4 Precision of CN2 with the different data management approaches for scenario A averaged over 10 trials with 20 batches each.

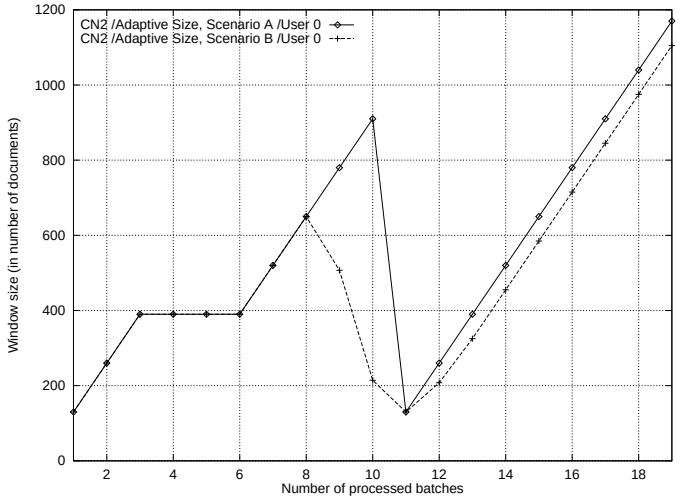

Figure 5 Adaptive window size for CN2 for scenario A and B averaged over 10 trials with 20 batches each.

Figure 5 shows the window size of the Adaptive Size approach in combination with CN2 over time on scenario A (and B). The Adaptive Size window grows up to its initial size of three batches (user-defined constant $|W_0|$) and keeps this size until the last of the initial batches for establishing the average values and standard errors (user-defined constant $M_0 = 5$). From the sixth batch on, the window adjustment becomes active and the window grows until the concept shift occurs in batch 10. Then the window is set to its minimal size of one batch, but starts growing again immediately afterwards, because no further shift or drift is detected.

4.4 Comparing Data Management Approaches with Complete Feedback on Scenario B (Concept Drift)

Like in table 5 for scenario A, the columns 2 to 5 of table 6 show accuracy, recall, and precision of all combinations of learning methods and data management approaches averaged over 10 trials with 20 batches each according to scenario B (table 3) with complete user feedback (user 0). The columns 6 to 9 contain results from additional experiments described later in section 4.5. Like in scenario A, using one of the two simple approaches with fixed window size instead of the Full Memory approach yields significant performance gains that can be further improved by the Adaptive Size approach, although the average improvement achieved by the window adjustments is smaller than in scenario A. As figure 6 shows for the example CN2, the three indicators work reliably in scenario B. Recall and precision again indicate the concept change much better than accuracy. The window size of the Adaptive Size approach with CN2 over time (figure 5) shows, that the window adjustment works in this scenario as well. The concept drift is already detected in batch 9 and the window size is reduced accordingly. The reduction of the window size continues until the end of the concept drift in batch 12. The fact, that the window was not radically set to its minimal size of one batch, shows, that the concept drift was not mistakenly suspected to be a concept shift.

		Full Memory User 0	No Memory User 0	Fixed Size User 0	Adaptive Size User 0	Adaptive Size User 1	Adaptive Size User 2	Adaptive Size User 3	Adaptive Size User 4
Naive Bayes	Accuracy	82.12%	92.47%	91.88%	92.72%	88.76%	85.13%	82.61%	75.53%
	Recall	66.02%	83.84%	83.37%	85.64%	74.35%	64.79%	65.72%	40.53%
	Precision	67.76%	86.00%	84.26%	85.69%	77.73%	73.02%	66.81%	46.03%
SVM	Accuracy	79.79%	90.88%	91.33%	92.01%	87.84%	83.80%	84.89%	76.61%
	Recall	49.33%	67.69%	73.76%	74.71%	68.32%	55.40%	70.89%	34.41%
	Precision	65.00%	89.22%	86.99%	88.83%	79.05%	74.19%	72.18%	46.85%
C4.5	Accuracy	78.29%	88.93%	89.48%	90.26%	85.29%	81.89%	81.74%	75.36%
	Recall	42.97%	68.12%	70.58%	72.34%	63.79%	53.74%	59.93%	34.84%
	Precision	49.41%	79.00%	81.07%	83.14%	74.90%	68.25%	65.66%	44.18%
CN2	Accuracy	78.94%	88.12%	88.51%	88.96%	84.26%	81.70%	77.58%	75.04%
	Recall	39.43%	58.59%	61.45%	62.43%	59.30%	45.38%	39.82%	23.52%
	Precision	62.28%	80.73%	84.17%	86.05%	72.72%	70.70%	53.02%	41.49%

Table 6 Accuracy, recall and precision of all learning methods combined with all data management approaches for scenario B averaged over 10 trials with 20 batches each.

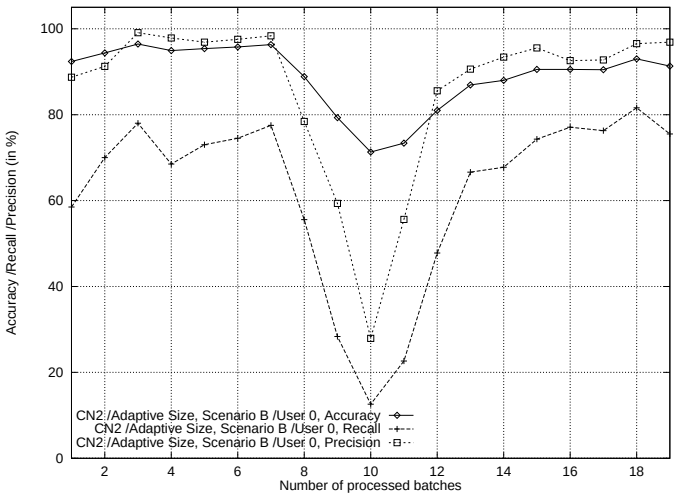

Figure 6 Accuracy, recall, and precision for CN2 with Adaptive Size for scenario B averaged over 10 trials with 20 batches each.

4.5 Results with Partial User Feedback (Scenarios A and B)

A second set of experiments was used to evaluate how reliable the indicators for concept changes and the window adjustment heuristic work with partial user feedback in order to test whether they are applicable to realistic filtering tasks. The Adaptive Size approach in combination with each of the four different learning methods was confronted with four different simulated user behaviours with only partial feedback (table 4) in both scenarios, A and B. As the plots of the window sizes for the different simulated users in figures 7 and 8 show for scenario A and B, respectively, for the example of CN2, the indicators still recognize the concept changes and the window size is adjusted accordingly for the users 1 to 3. For user 4 the recognition and adaptation only partially work, which can only hardly be seen in the graphs due to the extremely small number of labeled examples in this case. The four rightmost columns of the tables 5 and 6 show the averaged performance results for all four simulated users for the scenarios A and B, respectively. As one would expect, the performance drops with smaller numbers of labeled training examples, but at least for the users 1 to 3 the achieved accuracy, recall, and precision rates can still be considered helpful for realistic filtering tasks.

5 Conclusions

This paper describes indicators for recognizing concept changes and uses some of them as a basis for a window adjustment heuristic that adapts the window size to the current extent of concept change. The experimental results show, that accuracy, recall, and precision are well suited as indicators for concept changes in text classification problems, and that recall and precision indicate concept changes more clearly than accuracy. Furthermore it could be observed that even using a very simple window of fixed size on the training data leads to significant performance improvements

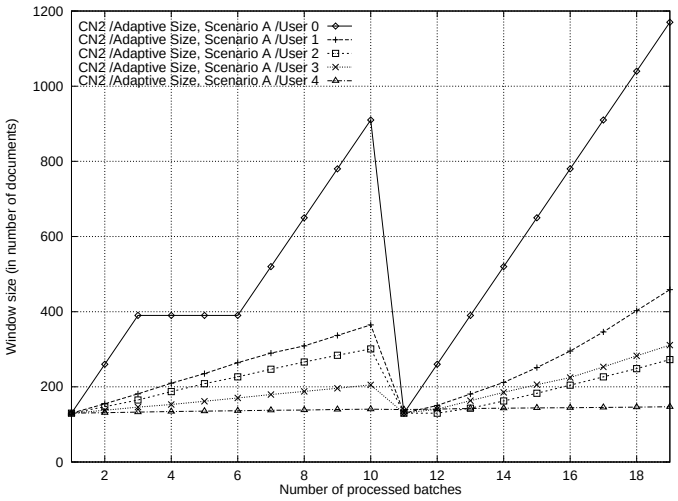

Figure 7 Window size for CN2 with the Adaptive Size approach for scenario A for the different user types averaged over 10 trials with 20 batches each.

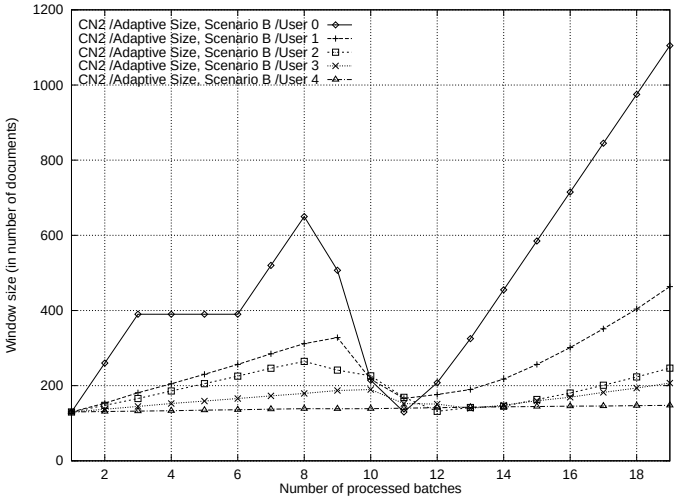

Figure 8 Window size for CN2 with the Adaptive Size approach for scenario B for the different user types averaged over 10 trials with 20 batches each.

for all tested learning methods compared to learning on all previously seen examples. The proposed adaptive window management approach yields further performance improvements over the best approach with a window of fixed size. The experiments with simulated partial user feedback demonstrated that the indicators for concept changes and the window adjustment heuristic still work even in most of these more realistic scenarios. Hence both, the indicators for concept changes as well as the window adjustment heuristic based on them, provide promising starting points for future research and applications in adaptive information filtering.

6 Outlook and Future Work

As partial user feedback leads to a smaller number of training examples, which in turn leads to a decrease of the performance of the filtering system, there is still room for improvement to let the performance drop more gracefully. A promising starting point are learning techniques able to learn from few labeled and possibly many unlabeled examples, especially because in most real-world information filtering tasks the unlabebeld examples by far outnumber the labeled examples and are usually much cheaper to get. Lanquillon describes a method for detecting concept changes without user feedback thereby reducing the need for user feedback to situations, where re-training the classifier becomes necessary [15]. Among the techniques for learning classifiers from labeled and unlabeled data are Co-Training [3] and the combination of Expectation Maximization (EM) with a naive Bayes classifier [17], which has already been successfully applied to text classification tasks.

The modification of the inductive learning task to the task of transduction [20] offers a further alternative for considering unlabeled examples. While for the inductive task general classification rules have to be learned for arbitrary test examples, for the transductive learning task only a classification model for an apriori known set of test examples needs to be learned. The unlabeled test instances can help to improve the performance of the learned classifier and/or to learn with fewer labeled examples, as has been successfully demonstrated in the text classification domain [8]. Obviously learning a classification model at each new batch in order to classify the newly received unlabeled examples can be regarded as a transductive learning task. The main objective for future work is the integration of techniques like the aforementioned to detect concept changes and adapt to them with as little user feedback as possible.

References

1. James Allan. Incremental relevance feedback for information filtering. In H. P. Frei, editor, *Proc. 19th Annual ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '96), Zürich, Swiss, August 18-22, 1996*, pages 270–278, New York, NY, USA, 1996. ACM Press.
2. Marko Balabanovic. An adaptive web page recommendation service. In W. L. Johnson, editor, *Proc. First Int'l Conf. on Autonomous Agents*, pages 378–385, New York, NY, USA, 1997. ACM Press.
3. Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In P. Bartlett and Y. Mansour, editors, *Proc. 11th Annual Conf. on Computational Learning Theory (COLT-98)*, pages 92–100, New York, NY, USA, 1998. ACM Press.
4. Peter Clark and Robin Boswell. Rule induction with CN2: Some recent improvements. In Y. Kodratoff, editor, *Machine Learning – Proc. Fifth European Conf. (EWSL '91)*, pages 151–163, Berlin, Germany, 1991. Springer.
5. Peter Clark and Tim Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.
6. William W. Cohen. Learning rules that classify e-mail. In *Proc. of the 1996 AAAI Spring Symposium on Machine Learning in Information Access (MLIA '96)*, Stanford, CA, 1996. AAAI Press.
7. Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Machine Learning: ECML-98, Proc. 10th European Conf. on Machine Learning*, Lecture Notes in Artificial Intelligence (LNAI 1398), pages 137–142, Berlin, 1998. Springer.
8. Thorsten Joachims. Transductive inference for text classification using support vector machines. In I. Bratko and S. Džeroski, editors, *Machine Learning – Proc. 16th Int'l Conf. (ICML '99)*, pages 200–209, San Francisco, CA, USA, 1999. Morgan Kaufmann.
9. Ralf Klinkenberg. Maschinelle Lernverfahren zum adaptiven Informationsfiltern bei sich verändernden Konzepten. Master thesis, Fachbereich Informatik, Universität Dortmund, Germany, February 1998.
10. Ralf Klinkenberg and Ingrid Renz. Adaptive information filtering: Learning in the presence of concept drifts. In M. Sahami, M. Craven, T. Joachims, and A. McCallum, editors, *Workshop Notes of the ICML/AAAI-98 Workshop* Learning for Text Categorization, pages 33–40, Menlo Park, CA, USA, 1998. AAAI Press.
11. Gerhard Kunisch. Anpassung und Evaluierung statistischer Lernverfahren zur Behandlung dynamischer Aspekte in Data Mining. Master thesis, Fachbereich Informatik, Universität Ulm, Germany, June 1996.
12. W. Lam, S. Mukhopadhyay, J. Mostafa, and M. Palakal. Detection of shifts in user interests for personalized information filtering. In H. P. Frei, editor, *Proc. 19th Annual ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '96), Zürich, Swiss, August 18-22, 1996*, pages 317–325, New York, NY, USA, August 1996. ACM Press.
13. Ken Lang. Newsweeder: Learning to filter netnews. In *Proc. of the 1995 Int'l Conf. on Machine Learning (ICML '95)*, 1995.
14. Carsten Lanquillon. Dynamic neural classification. Master thesis, Fachbereich Informatik, Universität Braunschweig, Germany, October 1997.
15. Carsten Lanquillon. Information filtering in changing domains. In T. Joachims, A. McCallum, M. Sahami, and L. Ungar, editors, *Proc. Workshop* Machine Learning for Information Filtering *held at* Int'l Joint Conf. on Artificial Intelligence (IJCAI-99), pages 41–48, Stockholm, Sweden, August 1999.
16. Tom Mitchell. *Machine Learning*. McGraw Hill, New York, NY, USA, 1997.
17. Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. Learning to classify text from labeled and unlabeled documents. In J. Mostow and C. Rich, editors, *Proc. Fifteenth National Conf. on Artificial Intelligence (AAAI-98)*, pages 792–799, Menlo Park, CA, USA, 1998. AAAI Press.
18. J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, USA, 1993.
19. Charles Taylor, Gholamreza Nakhaeizadeh, and Carsten Lanquillon. Structural change and classification. In G. Nakhaeizadeh, I. Bruha, and C. Taylor, editors, *Workshop Notes on Dynamically Changing Domains: Theory Revision and Context Dependence Issues, 9th European Conf. on Machine Learning (ECML '97), Prague, Czech Republic*, pages 67–78, April 1997.
20. Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, USA, 1998.
21. Georg Veltmann. Einsatz eines Multiagentensystems zur Erstellung eines persönlichen Pressespiegels. Master thesis, Fachbereich Informatik, Universität Dortmund, Germany, May 1997.
22. Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(2):69–101, 1996.