Handling Sudden Concept Drift in Enron Messages Data Stream

Miłosz R. Kmieciak and Jerzy Stefanowski

Institute of Computing Sciences, Poznan University of Technology, Piotrowo 2,
60-965 Poznan, Poland, milosz.kmieciak@cs.put.poznan.pl,
jerzy.stefanowski@cs.put.poznan.pl

Abstract. Detecting changes of concept definitions in data streams and adapting classifiers to them is studied in this paper. Many previous research assume that examples in a data stream are always labeled. As it may be difficult to satisfy in practice, we introduce an approach that detects a concept drift in unlabeled data and retrain a classifier using a limited number of labeled examples. The usefulness of this approach is evaluated in the experimental study with Enron data concerning classification of user's emails to multiple folders. First we show that these data streams are characterized by frequent sudden changes of concepts and that our approach can detect them. Results of the next comparative study demonstrate that our approach leads to the classification accuracy comparable to the periodic retraining of the classifier based on windowing, also reducing the number of examples to be labeled.

1 Introduction

Classification is one of the main tasks in machine learning and data mining. Many research have been undertaken and a great number of methods, based on different principles, have already been proposed. Most of these research concerns *static environments*, where a fixed set of labeled examples is presented to the learning algorithm which aims at producing the most efficient classifier with respect to the classification accuracy. However, in many real situations data sources generate rather continuous and time changing data, so learning is not a static task anymore. In particular it concerns evolving *data streams* where the data distributions and target class definition change over time [1].

A non–stable definition of classes in incoming data is known in literature as the *concept drift* [2]. Changes in data may lead to more or less sudden changes in class definitions. Typical literature examples of concept drift are weather predictions (that vary with seasons) or customers' buying preferences depending on, the so called, hidden context not given explicitly in features. As the reason of changes is hidden, not known a priori and not predictable with confidence, the learning task becomes very difficult [3].

Learning classifiers from data streams in presence of a concept drift is becoming important data mining problem also because of new types of applications as e.g.: analysing large sensor / measurement networks, monitoring telecommunication systems, traffic management, controlling production, classification of news, documents, computing in ubiquitous environments; for more details see [4].

In the last decade, learning and adapting to concept drift has been receiving a growing interest; for a review of methods see, e.g., [3, 5]. Most of these methods make an assumption that incoming data are always labeled (i.e. true class label is known for each example) and can be immediately used for learning classifiers. However, for many applications this assumption may be unrealistic or impractical as the class labels of newly coming examples in data streams are not "immediately" available but their obtaining is costly and need substantial efforts usually from human experts [6]. In our opinion, more realistic situation would be to start with a set of training examples for building the first classifier, next to process the stream of unlabeled examples with detecting possible concept drift. While the change is identified, a limited number of class labels should be accessed to retrain the classifier.

Partly inspired by the paper [6] we have decided to adapt the idea of using the decision tree to classify unlabeled examples in the data stream, and to model probability distribution in the stream basing on assignments of these examples to leaves of the tree. We propose to detect the concept drift as a result of discovering an increasing trend of differences between probability distributions in leaves - we introduce a technique for this identification. Moreover, we will use this approach for detecting the sudden concept drift (i.e. detecting new classes in data), which has not been originally considered in [6]. Presenting our approach and its experimental verification is the first aim of our paper.

We plan to carry these experiments with data sets concerning *folder categorization* problem, i.e. classification of a large number of emails into user-specific folders in their mailboxes. We will use Enron corpora of real email messages [7, 8] and try to learn how to automatically classify – assign emails to user's folders (class labels). We think that such receiving emails can be perceived as a stream data with its temporal nature and continuous evolution – as the time stamps of messages are passing by, some folders becomes less frequently used and new ones are created by the user when new threads / topics of messages are appearing. According to our best knowledge the Enron data sets have not been yet explored with respect to the concept drift. Therefore, preparing such a data set for experiments is the other aim of our paper.

2 Related Works

Usually two kinds of changes in classes (concepts) are distinguished in the literature: *sudden* (abrupt) and *gradual* concept drift [3].

The first type includes changes in the *class distribution* – it occurs when examples of the new class appear or examples of the already known class are not longer present in the stream. It directly influences the performance of the classification abilities as once generated classifier have been trained on different class distribution.

The other type of drift is not so radical, hence the differences in classification of examples can be noticed while looking in a longer period of time. Let $c(x)$ be the class produced by the classifier, the gradual drift appears if for the same examples x appearing in two different time moments t_1, t_2, the inequality $c_{t_1}(x) \neq c_{t_2}(x)$ holds. Zliobaite names it also as incremental (stepwise) drift [5].

Some authors study also *recurrent concepts*, i.e. previously active concepts may reappear after some time. This is not a completely exhaustive discussion of drift types – the reader is referred to [3, 2, 5] for more information on class label swaps or changes in underlying data distributions. Finally, a key issue while handling concept drift is not responding to minor fluctuations which can be perceived as noise.

Several techniques for handling concept drift have been proposed in the literature. Following the taxonomy from [3] we can distinguish three following groups of approaches based on: examples selection, weighting of examples or adaptive ensembles. Slightly different taxonomies of approaches are also presented in [1, 5].

Here, we focus mainly of the first group as it will be used in our experiments. The most common technique for selecting examples is based on periodic forgetting the older data and using the newest of incoming examples to retrain classifier. The simplest strategy is *sliding window* that moves over arriving examples – only arriving data are included in the current window. The best example of windowing is a family of FLORA algorithms [2]. Some techniques use windows of a *fixed size* which involves a problem of choosing a proper size (larger size is more useful for slower concept drift, while fails whenever drift suddenly occurs). Similar technique, also common in adaptive ensembles, is to divide data into non overlapping blocks (so called *data chunks*) and consider updating classifiers when a new block is available. Other adaptive solutions to non–fixed windowing based on heuristic adjusting the window size were also proposed; for a comprehensive review see [3, 5]. Due to ability to forgetting too old concepts all these techniques are treated as quite suitable for sudden and at least partly incremental drifts – however they do not detect *directly* changes.

Other approaches use direct detection of changes in data. After a drift is detecting older data are removed and a classifier is updated. However, the most widely used such direct approaches (called *triggers*) are based on observing decrease in the classification accuracy, which requires access to labeled stream of examples.

Some authors questioning the access to all labels, consider diving data into blocks when only a small fraction of examples is labeled – they often combine learning ensembles with semi–supervised clustering to handle the rest of unlabeled examples [9]. Yet other proposals [6] include active learning where a specific uncertainty sampling algorithm is used to select from the unlabled stream the limited number of most informative examples to obtain labels from an oracle.

Kuncheva in [1] discusses methods for signaling concept drift from unlabeled data, which are mainly based on monitoring probability distribution. This monitoring is similar to the novelty detection in data mining. Assuming a certain model associated with probability distribution, the probabilities for the current object x are calculated and compared to the model. If the differences between distributions are too high, the new object is not classified and added to the set of novel examples. When the number of novel examples reaches a certain level, the system should either stop classifying new objects or a new portion of labeled examples is used to retrain the classifier. These approaches requires proper estimation of probabilities. Klinkenberg et al. also draw attention to other indicators of changes [10] – for instance, when a classifier structure evolves in a new "direction" (like new rules in rule–based classifier) it may sign concept drifts.

3 Our Framework for Detecting Concept Drift

Taking into consideration the stream of flowing data and aim of direct detecting the concept drift from unlabeled data, we followed some inspirations from works on the stream oriented, *demand driven* methods proposed first in [6]. Originally it was more oriented to active learning in drift detection to obtain labels from the expert oracle. In our approach we are focused on early monitoring and drift detection methods, and we decided to adapt the idea of studying changes of data distribution by the classifier. Generally speaking our framework consists of the following steps:

1. Induce an initial classifier (in our case the decision tree) using the first *training_set_size* labeled examples in the stream.
2. Apply the most recent classifier to classify succeeding examples in the unlabeled stream.
3. Simultaneously, use the detection method to check possible concept drift. This step consists in modeling of data distribution in the leaves of the tree and comparing the current situation to the reference distribution. When the concept drift is detected, then perform the following actions:
   (a) Construct a new training set containing the *training_set_size* number of examples and get labels for them.
   (b) Remove an existing classifier and induce a new one, using training set from the previous step.

In our experimental study we consider two options for constructing training set in step 3a. The first one, called *extending previous set*, is inspired by a literature proposal of a landmark window, where the training set is composed of all passed data between a specific time stamp called landmark and the present [4]. As we set landmark at the very beginning of the stream, the induced model incorporates both the very old and the latest data. In the second option, the training set includes only *training_set_size* of the latest examples, thus limiting stream's history and causing classifier to "forget" the old concepts and focus on the new ones only.

Let us also notice that as the concept drift detection may be to early identified, actions in steps 3a and 3b of our framework are delayed for a number of *delay_period* examples in the incoming stream, in order to reflect sufficiently the new concept in the classification model.

To detect possible concept drift we observe changes in probability distribution of the unlabeled examples in the stream. To model probability distribution of the stream we use a decision tree[1]and register the assignments of the incoming examples to particular leaves. The key element of our approach is just observing changes in the data distribution according to *leaf statistic*. They express how examples occurring in the stream are spread in the attribute space, according to the current decision tree. The probability $P(x)$ of the example x can be approximated by the distribution of examples among the decision tree's leafs. Denoting the number of examples covered by the leaf l by n_l, the leaf statistic is given by the formula:

[1] Our implementation bases on the $J4.8$ decision tree algorithm from WEKA framework.

$$P(l) = \frac{n_l}{N}, \quad \sum_l P(l) = 1 \tag{1}$$

were N is the number of processed examples.

Any significant change of the leaf statistic value may indicate a concept drift. Therefore, we compare the current distribution of $P(l)$ with the reference value computed on the latest training set. The distance between them could be calculated in different way. In the current implementation we used the simple L_1 norm

$$d_P = \frac{\sum_l |P(l) - P_{ref}(l)|}{2}. \tag{2}$$

Our earlier empirical studies [11] show that using a simple threshold based method for monitoring distance statistic d_P may remain insufficient. Sudden concept drifts (like emerging new classes) influence statistic locally, causing rather trend changes than exceeding predefined threshold. Therefore, our drift detection methods analyses the variability on the distance value d_P, monitoring whether this value grows. We propose two options of this analysis, both basing on the limited horizon of latests $\delta = 30$ statistic's values. In the first approach, we count the number α of values forming the monotonic increasing sequence of values. Hence, if α/δ ratio exceeds given threshold $\beta = \lfloor 0.7 * \delta \rfloor$, an increasing trend is assumed. The second solution is based on the simple linear regression coefficient estimation, using the least squares method. This approach allows to observe trend sloper of the distance statistic, causing drift alarm on a given threshold.

Let us notice, that the described detection method is based on monitoring incrementally appearing examples so it could gain on its quality with growing number of processed data points. Therefore, step 3 of the framework is enabled after first $\gamma = \lfloor training\_set\_size * 0.8 \rfloor$ examples appear.

Finally, we mention the related work [12] where another indicator coming from leaf statistic was considered – an estimation of expected error (loss).

4 Pre-Processing of Enron Data Set

As a case study of mining data streams we chose a task of an automatic categorization of incoming email messages into multiple folders. In Section 1 we explained reasons for our choice, mainly coming from temporal nature with respect to time stamps of successive emails and continuous evolution in assignment of emails to particular folders (also with creating new folders if a new topic of messages appears). From the application point of view solving such a task could partly support users in filtering too many incoming emails and organizing them in a structure corresponding to different user's topics of interest. From the research perspective the interest is in machine learning methods for creating accurate classifiers on the basis of examples of user's previous decisions. Let us remark that *folder categorization* is a more general problem than an identification of spam only [7]. It is also a challenge itself as unlike traditional text categorization email messages are poorly structured comparing to longer texts and are written in an informal way.

Most of the empirical studies on the folder categorization rely on the self–gathered data sets. The most well known benchmark is the *Enron corpora* of real emails. It was made available to the researchers in 2002 as a result of the Enron Corporation scandal. This data was prepared by W. Cohen as a zip file repository containing 150 mailboxes of some company's management employees with more than 500000 messages[2]. An overview of the data set structure can be found in [8]. Because of the page limit we reduce discussion on related works, see e.g. reviews in [7, 8]. Shortly speaking, current researches are focused on evaluating the accuracy of various classifiers created by learning approaches with indication to Naïve Bayes, support vector machines, k-nearest neighbor or boosted decision trees. In some works, authors already considered the chronological order of email messages, however in the simplest way splitting each box into two (or a few) sets: training and testing [8]. To sum up, there were no research on handling concept drift in the perspective we consider in the following paper.

First of all, similarly to recommendations from [7, 8] we need to preprocess this data set. Many email boxes contain only very general folders like inbox, sent_mail, etc. As they do not refer to topic of user's interests we decided to remove these folders from consideration. After this step, we deleted 32 empty boxes and discovered that next over 45% boxes still contained folders with less 6 messages or the total number of remaining emails was smaller than 54. In some of other boxes, we identified a single folder which highly dominated the rest with respect to the number of messages. As in our experiments we plan to focus on the concept drift, we want to reduce the influence of other difficult aspects as imbalance. So we decided to finally choose 7 boxes with approximately balanced multiple folders with the possible high number of examples. The characteristics of chosen data sets are given in Table 1.

Name of data set	Folders no.	Messages no.	Folder size (no. of messages)		
			Minimum	Maximum	Average
farmer-d	13	2339	23	609	178
germany-c	8	939	31	390	117
lokay-m	7	1291	51	407	184
mann-k	17	1584	20	227	93
mann-k-mod	16	1357	20	186	84
rogers-b	9	877	29	235	97

Table 1. Basic statistics of the selected email boxes.

The next step consisted in transforming content of the email messages into an attribute–value representation suitable for learning classifiers. Each email was a text file including two major sections: header and body. From the header we extracted information concerning the following elements: *Subject*, sender information (*From, Replay to*), recipients (*To, CC, Bcc*), parameters describing the format of the message and *Data* – time and data when the email was send. These elements were parsed to get the complete email addresses or nicknames and were further treated as term – attributes.

Both subject field and the body of the message were processing as texts to find terms in the standard vector space model. We used typical tokenization and then eliminate the

[2] See at http://www.cs.cmu.edu/˜enron/

stopwords. As the number of identified terms was still too high (between 6007 and 11875) we used feature selection method based on Gain Ratio evaluation measure (from WEKA framework) and reduced them till 3400 for each data set (considering available memory resources and WEKA parameters it was sufficient for processing data). Although this number was still high, we did not decrease it as in the further experiment we planned to use the decision tree algorithm *J*4.8, which has an ability to select the limited number of the most important attributes to the tree.

5 Experimental Evaluation

There are two aims of our experimental studies with Enron data sets: (1) verification of the sudden concept drift characteristic of the data; (2) assuming that it exist we want to compare usefulness of two different approaches to handle it, including our approach described in Section 3.

Considering the first aim, we repeat our initial hypothesis that in the folder categorization we could expect changes in the class distribution in the stream of incoming messages. It is reasonable to expect new classes appearance when the user creates new folders, as well as situation when old class' members are no longer in the incoming email stream. As a verification, we create the *time–spatial diagram* visualizing the distribution changes in the time aspect. Let us discuss the diagram of the mann-k-mod dataset (see Figure 1). All the 16 classes in this data set are marked vertically (however we do not provide their labels for the sake of readability), whereas examples from the same class are plotted horizontally, from left to right according to the increasing time stamp order.

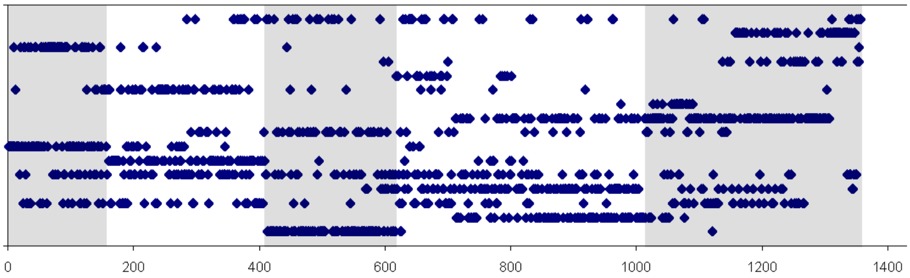

Fig. 1. The time–spatial diagram of classes' distribution of the mann-k-mod dataset, according to the message time stamp order increasing from left to right. Five manually created partitions are marked by shades of the background.

One can notice that changes occur suddenly rather than in the gradual fashion. In the figure we can identify time periods when the data distribution (and hence the target classes – folders) remain stable and the number of classes in the current stream unchanged. This kind of the *temporal locality* recognition seems to be important for the overall classification performance, as enables drift resistant mining techniques. For instance, when data distribution is unchanged, a larger time horizon can be taken into account, providing more confident stream processing. In Figure 1 we marked five partitions of the data stream, which may comply with the temporal locality of the data. In our opinion it clearly shows the concept drift presence in the studied data, with sudden changes in the class distributions between the partitions. Time plots of the remaining considered data sets show similar characteristic, with at least four local partitions each. Due to the page limit we skip their presentation and refer the reader to [11].

The moments of the sudden concept drift found in the Enron data sets directly influence classification of the new coming examples, as classifiers induced on the passed data may not able to correctly classify new data. Let us come back to Figure 1 diagram and the mann-k-mod data set, where we see five local partitions representing five rather different sets of target concepts. Let us consider the case when classifier was induced from the *training_set_size* = 200 first examples composing approximately the first partition. Figure 2 shows a classification accuracy of this classifier applied to the next examples in the stream. The value of this accuracy is updated after classifying each subsequent example of the stream, and hence it is referred to as a *cumulative accuracy measure*. We can notice that this value stabilizes after first 50 test examples at time stamp around 250 and remains in the range of 30–35% for next 150 data points. Then, the accuracy value decreases asymptotically since time stamp 400. Comparing this result with the time–spatial diagram from Figure 1 we can say, that the accuracy fall directly maps to the crucial moment of the sudden concept drift and the new temporal locality period. New email classes obviously cannot be predicted by the current classifier, thus decreasing the accuracy score.

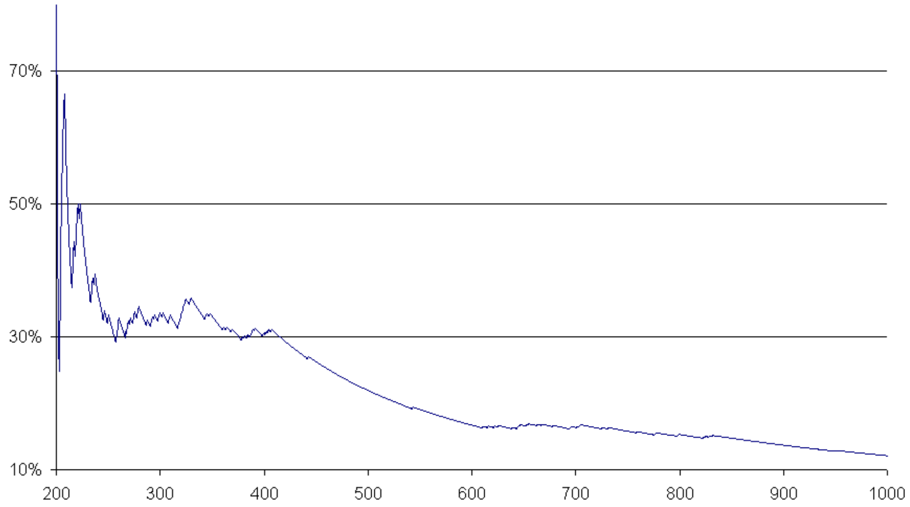

Fig. 2. The cumulative classification accuracy plot for the hold out evaluation scheme applied on the mann-k-mod dataset. The training set contains the $k = 200$ first examples. Each value of classification accuracy corresponds to the particular time stamp number of the test example.

Our next step is to check, whether this sudden concept drift can be efficiently detected by the drift indicator basing on the leaf statistic distance measure, as defined in Section 3. We compare plot shapes of the leaf statistic's values and the cumulative accuracy measure. Figure 3 shows relation between these values for mann-k-mod data set. Again, we inform that similar plots are observed for different moments of the data stream's flow as well as for the remaining data sets. Let us comment this figure. At the beginning, a certain number of examples from the processed stream is needed to stabilize these values. However later on, it can be noticed, that the leaf statistics indicator does react to the sudden concept drift observed in the data (the increasing trend of indicator is clearly visible), as it occurs around time stamp 400. The hyperbolic shape of this indicator seems to be complementary to the asymptotic cumulative accuracy fall. For comparative point of view we also plot the behavior of the expected error (loss) indicator from [12]. In our opinion the leaf statistic more strongly shows the increasing trend than the expected error and at least for the considered data sets is a better indicator of the sudden concept drift.

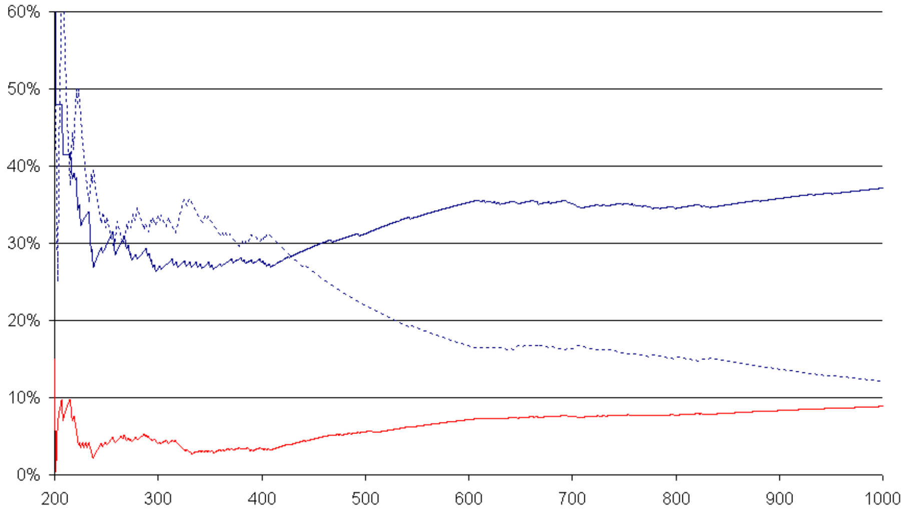

Fig. 3. The relation between the cumulative accuracy measure (dashed line), leaf statistic value (solid blue line) and the expected error rate (solid red line) in the context of passing stream examples. The examples' timestamps increase from left to right. Notice, that presented values of both indicators are from range $[0, 1]$.

The second aim of our experiment is to compare two different approaches to concept drift handling in the considered Enron data sets. The first approach is our direct drift detection as described in Section 3, with two options for constructing the training sets. As the second method, we chose a common method based on *fixed periodic updates* of the classifier without change monitoring. Also here, two options of creating a training set with labeled examples are considered. We call them *landmark window* and *sliding window* following inspirations from the literature [4]. However, unlike the incremental learning approaches to windowing where the classifier could be updated after reading a single example (see e.g. [2]), here we decided to split the stream into equal width blocks (similar to data chunks) and to learn a new classifier when the examples from the new block arrive. Thus, in the first landmark option a window – block of examples needs to be labeled and then it is added to earlier blocks while in the other option only the the latest block of examples is used to retrain the classifier. In our experiments we decided to fix both the period's length and the training set size. All approaches were implemented in Java using classes from WEKA environment. We based on Quinlan's tree induction algorithm, $J4.8$ classifier.

The evaluation scheme for both approaches is as follows. Once the model is induced basing on the chosen training set, it classifies sequentially arriving data points. The *cumulative classification accuracy* measure is updated on each single data point, reaching the final value representing to whole evaluation process score. In our opinion, this cumulative evaluation is an appropriate method for the stream algorithms, enabling both on–line verification and final summary. Besides the cumulative accuracy we analyze the number of retraining phases, as it directly refers to the expert's supervision for labeling examples.

The results of comparative experiments are given Tables 2 and 3. The best classification accuracy is always marked in bold. Let us discuss these results. For windowing approach, both versions achieve rather quite similar classification performance. However, better classification accuracies are obtained for sliding windows with the short period. We can say that the evolving nature of the stream in the considered Enron data sets favors more often model's update, determined by the period length (block / window size) parameter. Here, value *training_set_size* = 100 examples seems to be the most accurate. It is somehow consistent with previously discovered frequent changes of classes in all data sets. On the other hand, often updates result in the higher computational cost, as well as requiring more explicitly labeled examples.

Data set name	Landmark window						Sliding window					
	Length: 100		Length: 200		Length: 300		Length: 100		Length: 200		Length: 300	
farmer-d	**74.5**	23	73.9	11	67.1	7	60.8	23	65.1	11	62.0	7
germany-c	54.2	9	46.9	4	39.0	3	**55.3**	9	47.0	4	39.4	3
lokay-m	65.2	12	61.8	6	55.9	4	**67.6**	12	61.7	6	60.5	4
mann-k	**45.7**	15	39.3	7	35.1	5	45.6	15	40.0	7	36.5	5
mann-k-mod	46.0	13	36.8	6	33.5	4	**50.3**	13	34.0	6	36.2	4
rogers-b	65.1	8	56.3	4	54.2	2	**67.3**	8	63.9	4	55.3	2

Table 2. Evaluation results of the *windowing methods* with different lengths of blocks. The total accuracy values for each data set (email box) are shown, along with the number of retraining phases.

Data set name	Extending previous set						Limited horizon set					
	Length: 100		Length: 200		Length: 300		Length: 100		Length: 200		Length: 300	
farmer-d	59.1	4	**64.5**	4	60.6	3	55.1	9	60.8	3	55.3	3
germany-c	34.5	1	37.7	2	35.4	2	**54.2**	6	44.2	4	38.1	2
lokay-m	53.6	1	56.0	2	52.6	1	53.6	1	**63.0**	5	52.6	1
mann-k	20.8	3	21.7	3	12.8	1	**43.3**	13	34.2	6	12.8	1
mann-k-mod	30.0	4	33.2	3	25.5	3	**51.4**	11	32.9	6	38.7	4
rogers-b	61.1	2	37.7	1	34.2	2	**66.2**	3	37.7	1	32.8	2

Table 3. Evaluation results of methods basing on the *drift detection*, for different sizes of the retraining set. The total accuracy values for each data set are shown, along with the number of retraining phases.

Considering results of our approach for the drift detection, we can say that also constructing the training set with the most recent data is the best choice. We can suspect that for such latest examples it is easier to model the distribution changes in the stream, and hence to detect the concept drift. This also shows that it is easier to track drift between the two subsequent blocks of data, rather than in the context of all passed examples (here accuracy is definitely too low). Comparing results to the windowing, we can say that depending on the data they are slightly worse only or comparable. Except data sets farmer-d or lokay-m the differences are really small (no more than 2%). However, we could stress that the number of retraining phases is much lower than for windowing, what was the main motivation for our approach. We calculated the number of labeled examples used by the detection based limited horizon classifier in comparison to all available examples in the given data set. For farmer-d or rogers-b data sets we need to label 27% and 34% of all examples. For the rest of data it varies in range of 64% and 82%. The periodic updates based method of sliding window, uses approximately 95% of available examples, while remaining 5% is a result of the evaluation method.

Finally let us remark on the number of detected concept drift moments and factors it is influenced by. First of all, the temporal locality time periods may contain imbalanced data distribution, which affects the accuracy score, as well as the leaf statistic measure indicator. In the result, it is harder to observe trend slope changes of the distance measure as it is not so clear enough and may be caused by the data noise. The second factor is that Enron data sets may also contain other types of concept drift, like gradual changes, hence harder to observe in the time–spatial diagrams we used. Moreover, the drift dynamic may be high, favoring smaller training sets and more frequent updates of the model. However smaller training sets make the classifier suited for the temporal locality period only, once again affecting the accuracy score and leaf statistic measure. The above mentioned reasons made us to carefully tune the detection parameters for two data sets: germany-c and mann-k-mod, as the predefined parameter values for all data sets (described in Section 3) were not adequate for these data.

6 Conclusions

Our paper concerns constructing accurate classifiers that can adapt to concept drift in data streams. The motivation for our research is that one cannot expect complete labeling for all incoming examples in the stream. We claim that it is more realistic to reduce demands for labeling to a relatively smaller part of examples. Thus, we have presented an approach for handling the sudden concept drift which follows the above requirement. It uses a decision tree classifier to monitor unlabeled data streams and to detect the concept drift basing on observing a trend of changes probability distribution in the leaf statistics. Once it is identified, the classifier is retrained using a block of a small portion of the latest examples which need to be labeled.

We have evaluated this approach in the experimental study with the folder categorization of Enron users’ email messages. Our contribution is to clearly demonstrate that these data can be considered as stream characterized by the sudden concept drift. Our analysis also shown that changes of folders are quite frequent. As the number of real data to study concept drifts is still limited (see discussion[3]), we add a new one to the research community.

In the next experiments with this data set we proved that the sudden concept drift can be well identified by our approach. The final comparative experiment showed that our approach led to accuracy comparable or slightly worse only to popular windowing methods. The slightly better classification performance of the periodic learning of the next classifiers with the shortest size of the data window may be more well suited to quite frequent changes of classes in Enron data and other their dynamic characteristics discussed in the previous section. However, our approach limits demands for labeling to approx. 60% of examples needed by the first method. In our opinion the cost of labeling is not negligible in most real life applications, so its reduction is worth of obtaining a bit smaller classification performance.

Future research could concern resigning from fixed size set of retraining examples and incorporating a more active method for selecting the most informative examples for updating the classifier. Moreover, we evaluate this approach on other data characterized by different types of gradual drifts.

References

[1] L. I. Kuncheva, "Classifier ensembles for changing environments," in *Proc. of the 5th Workshop on Multiple Classifier Systems*, pp. 1–15, Springer LNCS vol. 3077, 2004.

[2] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, vol. 23, no. 1, pp. 69–101, 1996.

[3] A. Tsymbal, "The problem of concept drift: definitions and related works," tech. rep., Dept. of Computer Science, Trinity College Dublin, 2004.

[4] J. Gama and M. M. Gaber, *Learning From Data Streams : Processing Techniques in Sensor Networks.* Springer, September 2007.

[5] I. Zliobaite, "Learning under concept drift: an overview," tech. rep., Faculty of Mathematics and Informatics, Vilnius University, 2009.

[6] W. Fan, Y. Huang, H. Wang, and P. S. Yu, "Active mining of data streams," in *Proceedings of SIAM International Conference on Data Mining*, pp. 457–416, 2004.

[7] R. Bekkerman, A. McCallum, and G. Huang, "Automatic categorization of email into folders: Benchmark experiments on Enron and SRI corpora," Tech. Rep. IR-418, Center of Intelligent Information Retrieval, UMass Amherst, 2004.

[8] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *Proceedings of the ECML 2004 Conference*, pp. 217–226, 2004.

[9] K. L. Woolam Clay, Masud Mohammad, "Lacking labels in the stream: Classifying evolving stream data with few lables," in *Proceedings of the ISMIS 2009 Conference*, pp. 552–562, Springer Verlag, 2009.

[10] R. Klinkenberg and I. Renz, "Adaptive information filtering: Learning in the presence of concept drifts," in *Workshop Notes of the ICML/AAAI-98 Workshop Learning for Text Categorization*, pp. 33–40, AAAI Press, 1998.

[11] M. R. Kmieciak, "Learning multiple classifiers from text streams," Master's thesis, Poznan University of Technology, Poznań, Poland, 2009.

[12] S. Huang, "An active learning method for mining time-changing data streams," in *Proceedings of the 2008 Int. Symposium on Intelligent Information Technology Application, IITA'08*, (Washington, DC, USA), pp. 548–552, IEEE Computer Society, 2008.