Control and Cybernetics

vol. **40** (2011) No. 3

Semi-supervised approach to handle sudden concept drift in Enron data*

by

Miłosz R. Kmieciak and Jerzy Stefanowski

Institute of Computing Sciences
Poznan University of Technology
Piotrowo 2, 60-965 Poznan, Poland,

Abstract: Detection of concept changes in incremental learning from data streams and classifier adaptation is studied in this paper. It is often assumed that all processed learning examples are always labeled, i.e. the class label is available for each example. As it may be difficult to satisfy this assumption in practice, in particular in case of data streams, we introduce an approach that detects concept drift in unlabeled data and retrains the classifier using a limited number of additionally labeled examples. The usefulness of this partly supervised approach is evaluated in the experimental study with the Enron data. This real life data set concerns classification of user's emails to multiple folders. Firstly, we show that the Enron data are characterized by frequent sudden changes of concepts. We also demonstrate that our approach can precisely detect these changes. Results of the next comparative study demonstrate that our approach leads to the classification accuracy comparable to two fully supervised methods: the periodic retraining of the classifier based on windowing and the trigger approach with the DDM supervised drift detection. However, our approach reduces the number of examples to be labeled. Furthermore, it requires less updates of retraining classifiers than windowing.

Keywords: concept drift, incremental learning of classifiers, email foldering, Enron data.

1. Introduction

Development of computer technology enables automatic gathering and storing huge volumes of data. Looking for novel, interesting and useful knowledge representation in these data is the aim of data mining which has been intensively developing since the 1990s (Klosgen and Zytkow, 2002). Most of previous and current research on data mining is devoted to *static environments*, where data

*Submitted: March 2011; Accepted: August 2011.

are stored in data warehouses or other special repositories and can be accessed several times, if needed, by algorithms. Furthermore, knowledge patterns, hidden in data, are rather of fixed, static nature.

However, a new class of emerging applications recently becomes more visible where data are continuously generated at a high rate in a form of *data streams*. A data stream is an ordered (either by arrival time or by timestamp) sequence of instances coming from *dynamic and time changing environments* (Gama and Gaber, 2007). Data stream applications often described in the literature include: performance measurements in networks, monitoring and traffic management, security, management of call records in telecommunications, analysis of log records generated by web servers, sensor networks.

Data streams are characterized by huge volumes of records (possibly infinite), arrival at a rapid rate, fast changes and frequent need for quick, real-time response or analysis. This implies new requirements for data mining algorithms such as constraints on the volume of memory used by algorithms, small processing time per record, single scan of incoming data and ability to handle time changing phenomena (for more details see, e.g., Domingos and Hulten, 2000; Gama, 2010). Most of well known data mining and previous machine learning approaches ignore the data stream characteristic and these requirements. Therefore, one can observe a growing research interest in new approaches to learn from *evolving data streams*.

In this paper we focus our attention on *supervised classification learning*, which is one of the most popular tasks in data mining, machine learning or pattern recognition (Bishop, 2007; Han and Kamber, 2006; Mitchell, 1997). It is typically defined as discovering from data (learning examples) knowledge about assigning objects, described by a fixed set of attributes, to one of the pre-defined classes. A large number of methods have already been proposed to construct accurate classifiers from static data. However, they fail to efficiently learn from evolving data. This is partly justified by the data characteristics mentioned above, but also by yet another distinctive feature of classification from evolving data – *concept drift*. Concept drift implies that class label of incoming examples may change over time, novel classes may appear in the stream, and previous class definitions may become no longer valid.

Typical literature examples of concept drifts are weather predictions (that vary with seasons), financial analysis of frauds where the source of concept drift is the evolving users' behavior, or customers' buying preferences depending on the so called hidden context not given explicitly in the available attribute descriptions. Other real life examples coming from monitoring systems, transportation systems, spam or text filtering, economics or biomedicine, analysing behaviour of user's visiting web services, managing company relations with customers, monitoring social networks are discussed in Gama and Gaber (2007), Gama (2010), Zliobaite (2009).

Let us repeat after Tsymbal (2004) that as the reason of changes is hidden, not known a priori and not predictable with confidence, the learning task

becomes very difficult. We can also refer to typical assumptions of learning algorithm where examples are described by a finite set of attributes. In evolving data this assumption is violated as the *context* of the problem changes, which means that some *hidden variables* (hidden for the learning algorithm) may change over time (Widmer and Kubat, 1996). As a result, target classes learned at one time moment can become inadequate in some new moments.

Another issue of changes is related to their rate. Usually two kinds of changes are studied in literature (see Gama, 2010; Kuncheva, 2004; Tsymbal, 2004; and Zliobaite, 2009) or *sudden concept drifts*. The former is associated with slower changes in the target concepts (e.g. changes in prices of some food), while the sudden concept drift refers to abrupt changes (i.e. novel classes may appear rapidly in incoming stream of examples).

Let us clearly stress that in our study we are particularly interested in sudden concept drift, whereas computational requirements, typical for huge data streams, are not the main aim of our study.

To illustrate the importance of detecting novel classes let us refer to the problem of network intrusion detection discussed in Woolam et al. (2009). The detection of unwanted computer access (attack or suspicious behavior) is of crucial importance in any real monitoring system. In this problem the source of the concept drift is mainly linked to the attackers, as the actions undertaken by them usually evolve with time, so as to take the computer security software by surprise. In particular, when a new kind of intrusion occurs, the system should not only be able to identify that it is an intrusion, but also that it is a new type of attack, to find a proper defense action against it or at least to alarm in advance the human experts as they need to analyze logs more intensively.

Although there exists some earlier research on *incremental learning* (see, e.g., Hulten et al., 2001; Kubat, 1989; Schlimmer and Granger, 1986), the typical methods are not efficient in handling concept drift and this decreases the overall prediction abilities of traditionally constructed classifiers. In particular, this refers to detecting new classes in the stream. Let us remind that traditional classifiers can correctly process examples of only these classes, which they have been trained on. When a new class appears in the stream, its examples will be misclassified. Moreover, its presence may make recognition of other classes more difficult. Therefore, research on new types of specific algorithms for handling concept drift is still a challenging problem.

In the last decade, learning in the presence of concept drift has been receiving a growing interest. Sliding window approaches model the process of forgetting older concepts with consecutive classifier updates. Other proposals include new online algorithms, special detection techniques or adaptive ensembles. For a review please refer to Tsymbal (2004), Zliobaite (2009).

Most of the proposed methods make an assumption that incoming learning examples are always labeled (i.e. true class label is known for each example) and can be immediately used for learning classifiers. For instance a reader can consult a newest comprehensive review in Gama (2010) where nearly all pre-

sented approaches adapting to concept drift evaluate their performance on the most recent examples with available class labels. Using machine learning terminology these are completely *supervised approaches* as in traditional approaches examples are labeled either manually by human experts or by an outside oracle. However, for many data streams applications this assumption may be unrealistic or impractical, as the class labels of newly coming examples in data streams are not "immediately" available. Their acquisition is costly and needs substantial efforts, usually of the human experts (Fan et al., 2004; Woolam et al., 2009; Masud et al., 2008). In particular cases, it may be impossible to access true class labels even with a small delay. For instance, in the financial fraud detection, information on fraud transactions is usually known after a long delay (e.g. when an account holder receives the monthly report, Fan et al., 2004). To sum up, in changing environments, when data appear quite quickly, it may not be possible to label (in particular by a human) all the examples, as they sequentially arrive. We could rather expect that only a small portion of data is eventually labeled and available to the learning algorithms, leaving other parts of data unlabeled. Therefore, in our opinion, more reasonable solution is to consider *semi-supervised* approaches to learn classifiers from streams, where algorithm processes only partially labeled examples in distinction from the traditional algorithms.

Our proposal of such a semi-supervised learning of concept drift is to generate an initial classifier, basing on a limited number of labeled training examples, and to use this classifier to process the upcoming stream of unlabeled examples, while simultaneously detecting possible concept drift. When a change is detected, a relatively limited number of class labels of the latest examples are acquired, to train a new classifier, applied again to classify the subsequent unlabeled examples.

Although our proposal resembles somehow the already known *trigger methods* (Zliobaite, 2009), where the change detector starts adaptation of the learner to data, we stress that these methods are completely supervised (as the most of well the known methods, DDM, Gama et al., 2004, or EDDM, Baena-Garcia et al., 2006); on-line detectors are also used in the following ensembles of classifiers, Nishida, 2008; Deckert, 2011, while change detection in unlabeled examples is definitely more difficult. It usually requires modeling and estimating specific probability distribution – for discussion see Kuncheva (2004). However, one could find a less complicated approach of *active mining of data streams* (Fan et al., 2004). The main element of this proposal is to detect changes by estimating special loss errors made by a tree classifier on the unlabeled examples. Inspired by this paper, we adapt the idea and model probability distribution of upcoming examples with statistics based on assignments between these examples and the tree leaves. Our contribution is also a different technique for detecting the concept drift as a result of discovering an increasing trend of differences between probability distributions in leaves. Moreover, we use this adapted approach for detecting the sudden concept drift, which has not been originally considered in Fan et al. (2004).

Furthermore, we would like to study experimentally the usefulness of this approach on a real life problem. A problem, which could be itself a non-trivial data mining task, with an intuitive meaning. Let us remind that several current experiments with concept drift methods, also the most related one from Fan et al. (2004), were carried out with artificial data. The reader could also refer to experiments with MOA framework, where examples of the most popular generators of such data sets are shown (Bifet et al., 2009; Bifet and Kirkby, 2009). We consider a case study of *folder categorization* basing on the *Enron corpora* of real email messages (Bekkerman et al., 2004; Klimt and Yang, 2004).

Folder categorization is a problem of classifying a large number of emails into user-specific mailbox folders. Previous research concerned the most accurate methods of automatic learning of classifiers with class labels corresponding to users folders. According to our best knowledge the Enron data sets have not been explored yet with respect to concept drift. Therefore, preparing such data sets for experiments is the other aim of our paper.

Besides showing the details of the proposed semi-supervised approach applied to Enron data, we compare it with two popular fully supervised methods. First method is based on sliding windows and the other uses the most popular DDM trigger method (Gama et al., 2004). Besides checking whether the new approach could lead to comparable classification accuracy, we want to verify whether it could reliably and precisely detect sudden changes of classes and quickly enough adapt to them. Particular attention will be also paid to evaluating the new approach requirements for labeling examples. Let us remark that the main motivation of our study is to give up the need for completely labeled stream of examples (which is a critical assumption of many existing approaches!). Thus, we will evaluate how much the new approach reduces the number of examples for labeling comparing to sliding windows.

This paper is organized as follows. The next section presents related work in concept drift detection, including previous semi-supervised approaches. In Section 3 we present our framework and detection method details. Section 4 describes the Enron data set and details of preparing it for the experiments are given in Section 5. The next section contains result of experiments and Section 7 concludes the paper.

2. Related work on concept drift

2.1. Taxonomy of concept drift

In the classification problems in non stationary environments the class distribution can change over time. In the literature, one can find at least two following formulations of concept drift. Kuncheva (2004) presents a probabilistic point of view, where concept drift may occur as a result of changes in one of the following probabilities: prior probability of classes, class-conditional or posterior probability. On the other hand, Zliobaite (2009) defines concept drift as

an unforeseen substitution of data source S_1 (with an underlying probability distribution Π_{S_1}) with another source S_1 (with probability distribution Π_{S_1}).

Two kinds of changes in classes, and hence two kinds of concept drift are distinguished in the literature: *sudden* (abrupt) and *gradual* (Tsymbal, 2004). The first type includes changes in *class distribution* – occurring when examples of a new class appear or examples of the already known class are not longer present in the stream. It directly influences 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 by looking over a longer period of time. Let $c(x)$ be the class produced by the classifier, 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 (2009) names it also the incremental (stepwise) drift. As in this paper we do not discuss drift types completely, reader is referred to Tsymbal (2004), Widmer and Kubat (1996), Zliobaite (2009) for more information on class label swaps, changes in underlying data distributions and reoccurring concepts.

2.2. Basic methods

Several techniques for handling concept drift have been proposed. Following the taxonomy from Tsymbal (2004) we can name three groups of approaches based on: example selection, weighting of examples or adaptive ensembles. Slightly different taxonomies are also presented in Kuncheva (2004). Moreover, one can also distinguish methods according to two aspects: when to adapt the classifier (spreading methods from triggers to evolving ensembles) and how to adapt to concept drift (which distinguishes mainly between example selection and parameterization of the base learning algorithm), Zliobaite (2009). Finally, a key issue while handling concept drift is to identify minor fluctuations as noise.

Here, we focus mainly on example selection and triggers, as they will be used in our experiments. The most common technique for selecting examples is based on periodic forgetting of the older data and using the newest of incoming examples to retrain the classifier. The simplest strategy is a *sliding window* that moves over arriving examples – only arriving data is included in the current window (see the family of FLORA algorithms from Widmer and Kubat, 1996). Some techniques use windows of *fixed size* which involve a problem of choosing a proper size (larger size is more useful for slower concept drift, failing whenever drift suddenly occurs). Other adaptive solutions to non–fixed windowing, mostly based on heuristic adjustments of the window size, were also proposed. Similar technique, also common in the adaptive ensembles, is to divide data into non overlapping blocks (so called *data chunks*) and consider updating classifiers (re-calculating weights of classifies in the ensemble, removing too weak components or learning new ones) when a new block is available. Examples of such block based ensembles are AWE or AUE, see Brzezinski and Stefanowski (2010) for their characteristics. For a review of many approaches, see, e.g., Gama (2010), Tsymbal (2004), Zliobaite (2009).

Let us notice that the above described approaches do not detect concept drift directly, but rather adapt to it mainly as a result of the window based example selection, or the consecutive classifier updates. Completely different approaches, called *triggers*, are based on direct detection of changes in data (Zliobaite, 2009). When drift is detected, then the classifier update should be triggered. A well known example of such trigger is the *drift detection method*, DDM, proposed by Gama et al. (2004), which controls the number of errors made by the classifier while processing the incoming examples. A significant increase of the error statistic, exceeding the predefined threshold, indicates that the class distribution is changing and the current classifier is inappropriate. Discussion of other proposals of direct detection methods, like EDDM (Baena-Garcia et al., 2006), and their experimental evaluation is presented in Bifet et al. (2009). However, let us stress that these methods also require access to the labeled stream of examples.

2.3. Unlabeled data and semi-supervised approaches

Full access to labels for fast arriving examples in the stream is questioned in the literature. This leads either to processing completely unlabeled data or to semi-supervised paradigms, where only a small fraction of training examples is labeled. Then, the learning algorithm benefits from generalizing only a limited number of labeled examples, while processing the incoming stream of unlabeled examples.

Generally speaking, tracking concept drift from unlabeled data could be based on monitoring distributions over two different time-windows. A reference window that should summarize some past information and a window over the most recent examples. Kifer et al. (2004) proposed statistical techniques or tests based on Chernoff bound to examine samples taken from these windows and decide whether two probability distributions are different. More details on comparing two distributions are given in Gama (2010).

Kuncheva (2004) discusses the methods for signaling concept drift from unlabeled data, which are mainly based on monitoring probability distribution. It is similar to the *novelty detection in data mining*. Assuming a certain model, associated with probability distribution, 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, but added to the set of novel examples. When the number of novel examples reaches a certain level, system should either stop classifying new objects or the classifier is retrained basing on a new portion of labeled examples. These approaches require proper estimation of probabilities. Again, other approaches to novelty detection are briefly discussed in chapter 9 of Gama (2010).

Klinkenberg and Renz (2998) also draw attention to other indicators of changes. For instance, when a classifier structure evolves in a new "direction" (like new rules in rule–based classifier) it may mean concept drifts.

Woolam et al. (2009) presented a kind of hybrid approach to process both labeled and unlabeled examples from stream divided into chunks (blocks). First, they introduced a semi-supervised clustering algorithm to find several clusters from the partially labeled examples. A summary of the statistics of examples belonging to each cluster (called a "micro-cluster") is used as a basis for the k-nearest neighbor classification algorithm, i.e. a new example is assigned to the most frequent class on the basis of labeled examples in the nearest micro-clusters. In order to follow the stream changes, authors propose a strategy of keeping an ensemble of such k-nn models. Similar approaches to partly labeled data blocks or detection of novel classes are also considered in Masud et al. (2008, 2009).

2.4. Demand-driven active mining of data streams

The approach of *active mining of data streams* from Fan et al. (2004) is most related to our proposal discussed in this paper due to sharing of motivations similar to ours that the assumption of mining completely labeled data can be hardly satisfied in practical situations. Even if it is possible to obtain the true labeling, it will be available after a certain period of time. Further, in such a case "it is a common practice to *passively* wait with refreshing the current classification model until the labeled data is available". However, delay between arrival of real data elements in the stream and providing labels for them might be quite long, causing the concept drift detection moment to overlook real drift occurrence.

This issue is addressed by the *demand-driven framework* proposed in Fan et al. (2004), as it might benefit from the early monitoring of changes in unlabeled data with cost saving active selection of informative examples, achieving the accuracy of well known drift detection methods. The framework can be summarized in the following three steps:

1. Basing on the unlabeled data, the first guess on possible concept drift is made. It is done by estimating the loss or error rate for the existing model that classifies this incoming unlabeled data.
2. When the estimated value exceeds the predefined application-specific thresholds, the algorithm selects the most informative unlabeled examples forming the query subset for obtaining their true labels. With these labels, the true error of the classifier is again estimated.
3. If the error estimated in step 2 is verified to be higher than the tolerable threshold, the old classifier is updated using the trained set (with acquired labels), constructed in the previous step.

Note that authors of the demand-driven framework proposal clearly state that they analysed the gradual concept drift[1]. In our opinion the key point of this approach is the idea of detecting concept drift by observing changes in

[1] "In our work we explicitly exclude new class labels (...)", Fan et al. (2004), Section 2.1.

decision tree, which was used to classify unlabeled examples. More precisely, it is proposed to analyse how consecutive stream examples are assigned to leaf nodes, as it allows to approximate the probability distribution of examples with respect to combination of attributes used in the tree paths. The approximation can be expressed by observable statistics in the decision tree, e.g. by leaf statistic described in Section 3. However, Fan et al. (2004) eventually propose to detect the drift effects on decision trees by means of special loss functions[2].

First, the *classification error* $err(l)$ of each leaf is approximated on the basis of examples in the stream that are expected to be misclassified, and hence the estimated overall classification error of the model is formulated as $\sum_l err(l)$. These values can be subsequently weighted by relative numbers of examples covered by the particular node. This current weighted overall error is compared against its reference value formerly calculated at the tree generation moment. If the difference exceeds the predefined threshold, the concept drift is expected. In the next step, conforming to the active mining demand-driven framework, a query subset of limited number of the most informative examples is formed. Taking this subset of examples, a true class label is queried and attached to each element. This could be done either by the specific random selection (Fan et al., 2004) or according to the paradigm of *active learning* selection of the most informative examples from the data stream. Huang (2008) proposed to use a variant of the *uncertainty sampling* in its heterogeneous form (originally introduced in (Lewis, 1995)). The active learning system is composed of the naive Bayes classifier that indicates the most uncertain examples included in the query set and the additional classifier, labelling all the remaining ones (Lewis, 1995). The main challenge of the selective sampling of examples is to choose the most informative ones — not only representing the current chunk of data, but also the subsequent chunks.

3. Our framework for detecting concept drift

Following motivations of direct detection of concept drift from unlabeled data and inspirations from the *active demand driven* method of (Fan et al. (2004) we decided in our approach to adapt their idea of studying changes of data distribution by the classifier. Thus, 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 incoming data.
2. Apply the most recent classifier to classify succeeding unlabeled examples.
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 concept drift is detected, then perform the following actions:

[2]0-1 loss function refers to the error rate in the statistical learning terminology.

(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 the training set from the previous step.

Let us stress that we need to construct tree classifiers from labeled data – though they should be learned from relatively small portion of examples comparing to the total size of the stream. The parameter *training_set_size* of the latest examples has been evaluated in our experiments, and 100 or 200 examples seems to be a sufficient number for effective training starting classifier. Then, up to 100 examples has proved to be sufficient for retraining a tree classifier when the possible concept drift is detected. Generally speaking, we promote the option when the training set includes only *training_set_size* of the latest examples, thus limiting stream history and causing classifier to "forget" old concepts and focus on the new ones only. This is similar to solutions considered in DDM like methods (Gama, 2010).

Let us also notice that as concept drift detection may be too 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 decided to observe changes directly in probability distribution of the unlabeled examples in the stream – which is also slightly different to Fan et al. (2004). Following inspirations from Fan et al. (2004) we use a decision tree[3] to model probability distribution and we 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*. This shows 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 leafs. Denoting the number of examples covered by the leaf l by n_l, the leaf statistic is given by the formula:

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

were N is the number of currently processed examples (i.e. starting from the moment of training the last classifier) and obviously $\sum_l P(l) = 1$.

If the combination of attribute values in the current part of the stream is different than this existing in the reference training set, it will be directly reflected in $P(l)$. Any significant change of the leaf statistic values may indicate 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 ways. In the current implementation we used

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

the simple L_1 norm

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

Let us briefly discuss computational requirements of the presented approach. One question is the tree size, as for the experiment we used a standard pruned option of the C4.5 tree (Quinlan, 1993), a polynomial heuristic that has been successfully applied in many data mining tasks (Han and Kamber, 2006; Klosgen and Zytkow, 2002; Mitchell, 1997). For the exemplary mann-k-mod dataset and a sliding window approach of $training\_set\_size = 200$ examples, the tree size varies from 35 to 79 with leaf number accounting for roughly 50% of that value. These values are rather small, especially in comparison to attributes space of 3400. Moreover, classification of a new incoming example is cheap, as it traverses the tree only once from root to a single leaf, passing through a limited number of attribute tests (see also discussions in Fan et al., 2004). The second interesting matter is a requirement concerning the number of retraining phases. However, as we show in further experimental evaluation, in case of the demand driven detection, it is at least 10 times less than in a simple sliding window technique.

Moreover, our earlier empirical studies (Kmieciak, 2009) show that using a simple threshold based method for monitoring distance statistic d_P as originally considered in Fan et al. (2004) may be insufficient. Sudden concept drifts (like emerging new classes) influence statistic locally, causing rather trend changes than exceeding predefined threshold. Moreover, real life data sets vary in the sense of thresholds that properly indicate the concept drift moments. Therefore, our drift detection method analyses the variability of the distance value d_P, monitoring whether this value grows. We propose two options of this analysis, both basing on the limited horizon of latest δ statistic values[4]. In the first approach, we count the number α of values forming the monotonic increasing sequence of values, i.e. the number of values greater than all previously seen in the current horizon. In our opinion this step could be modified, relaxing the monotonic constraint, by observing a number of values greater than maximum observed in the current horizon. Furthermore, if α/δ ratio exceeds given threshold $\beta = \lfloor 0.7 * \delta \rfloor$, an increasing trend is assumed, as the number of statistic values that compose a strictly increasing sequence is satisfying. In other words, distance value d_P has been increasing for at least $\beta * \delta$ observations. Another solution we studied is based on the simple linear regression coefficient estimation, using the least squares method. This approach allows to directly observe trend slope of the distance statistic, causing drift alarm at a given threshold. The main issue related to trend slope observation in the real life data, however, is normalization. Our empirical study shows that alarm threshold may significantly differ between data sets, and hence needs to be preset experimentally. We agree that this is a difficult requirement for analysing larger data streams.

[4]In experiments we tuned $\delta = 30$.

Let us note, that the described detection method is based on monitoring incrementally appearing examples so it could improve 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. This may affect the overall classification performance, as disabled detection step might miss concept drift moment. Please refer to Section 6 for the experimental evaluation. Let us also notice that provided parameter values have been set basing on experiments, and this issue may be potentially addressed by further study.

4. Folder categorization: a case study of Enron data

As a case study of incremental learning from data stream with sudden concept drift we chose an *email foldering task*. This is one of the email classification problems and consists in assigning the incoming email messages into user defined folders. The foldering task is often referred to in the literature as a *folder categorization*. It is one of these text classification problems, which have been receiving a research interest in the last decades as the machine learning applications (Bekkerman et al., 2004; Clark et al., 2003; Sebastiani, 2002). However, it is a different task than definitely more popular identification of unsolicited emails (spam detection). Here, one is interested in automatic learning to assign, usually already filtered non-spam messages to user's *multiple* folders based on examples of the previous user's strategy. Solving such a task could support users in filtering too many incoming emails and organizing them in a structure corresponding to different topics of interest (Clark et al., 2003).

The importance of email communication has increased dramatically and many users are receiving too many emails everyday. Browsing them and answering to the most crucial emails has become increasingly difficult and time consuming. So, the need to organize emails has grown as well. Modern email software often provides simple tools for filtering incoming messages, based on keyword-containment rules specified by users. Manual definition of such rules and their tuning is rather difficult. So, it is challenging to check whether one can apply machine learning methods to assign messages into user folders based on the examples of previous classification decisions (Clark et al., 2003). Of course, in case of concept drift we do not mean completely automatic tools. We mean, instead, a semi-automatic philosophy with an interaction between a tool and an experienced user. In case of detection of some changes or appearance of a new possible label, the system should inform a user by a kind of warning and asks for reaction (e.g. checking whether it is necessary to create a new folder, etc.). Such tools could be learning and working in the spirit of rather *active learning* with sending some actions / questions to users.

This problem of folder categorization is still an under-explored research field compared to spam filtering, press news categorization, etc., see, e.g., discussions in Bekkerman et al. (2004). Note that the nature of email classification is different from traditional text categorization, which mainly involves quite well

structured text, as e.g. news or medical articles (Sebastiani, 2002). First of all, an email message is rather very short compared to a typical text. Then, the contents of messages (bodies) are poorly structured and are written in an informal way. There is no standard way of processing them to one format although several authors showed that besides creating term representation of the body of the text and also the email subject, it is useful to analyse some other fields from the header of the message (see discussions in Klimt and Yang, 2004; Stefanowski and Zienkowicz, 2006). Moreover, a single folder may represent many email discussion threads which regard the same general subject or they are connected in some other way. It may be connected with time dependent characteristics as some email messages (and user's decision on its assignment to folders) only make sense in the context of previous ones. However, the similarity between threads remain not obvious for all threads, especially when one considers the common terms (keywords in bag-of-words representation) they contain. Let us also notice that email folders do not necessarily correspond to simple semantic topics, sometimes they correspond to unfinished projects, groups of various interests, certain recipients or loose agglomerations of topics (Bekkerman et al., 2004). Email content and foldering habits usually differ from one user to another, so automatic tools should be personalized, as they may perform well for one user but they may fail for another at the same time.

To sum up, we claim that it is a challenging research problem. Up to, now research concerned mainly problems of finding proper representations of emails and studying the predictive abilities of various learning algorithms (see, e.g., discussion in Bekkerman et al., 2004). In Section 1 we also explained other reasons for our choice, mainly coming from temporal nature of an email stream with respect to time stamps of successive emails and continuous evolution in assignment of emails to particular folders – users create new folders if a new topic of messages appears and let other folders fall out of use.

Let us notice that most of empirical studies on folder categorization rely on the self–gathered data sets very often taken from private mboxes of researchers or their students (see e.g. Stefanowski and Zienkowicz, 2006). Due to privacy of personal correspondence such data sets were not made available for public access. Several benchmark data sets are publicly available, though, for the classical text classification, like Reuters news collections. The situation changed when during investigations on the Enron Corporation scandal the Federal Energy Regulatory Commission made public the content of mboxes of some employees from this corporation. This quite large collections of emails was noticed by researchers and became the most known and popular benchmark for classifying emails. This data was prepared by W. Cohen as a zip file repository containing 150 mailboxes of employees with more than 500,000 messages[5]. An overview of the data set structure can be found in Klimt and Yang (2004). For a broader discussion on related works, see, e.g., reviews in Bekkerman et al. (2004), Klimt

[5]See at http://www.cs.cmu.edu/~enron/

and Yang (2004). Shortly speaking, research focuses now 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. Some 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 (Klimt and Yang, 2004). Stefanowski and Szopka noticed earlier that sliding windows significantly improved classification accuracy. However, as they focused on choosing the most accurate classifier and feature selection they did not analyse more deeply concept drift (Szopka, 2007). So, there has been no research on handling concept drift in the perspective we consider in this paper.

5. Pre-processing of Enron data

Similarly as Bekkerman et al. (2004), Klimt and Yang (2004) we state that the Enron data require preprocessing. This includes selecting the most interesting mboxes according to learning task being studied, here: concept drift. Moreover, folder content has to be "cleaned" and finally proper attribute-value representation constructed.

This pre-processing phase is mainly needed to construct new "benchmark" data sets for testing various algorithms. Therefore, pre-processing is not a part of our approach but just an independent task.

5.1. Choosing mailboxes and folders

The available Enron zip file is divided into directories, each corresponding to the particular employee's email box and containing messages put into categories-related subfolders. However, since this structure is taken directly from the personal email client applications, it contains redundancy, and needs to be processed prior to any further analysis. Very often, email boxes from the Enron dataset contain general folders including note messages or sent/received ones. As they do not refer to main topic of user's interests we decided to remove these folders from consideration.

Another issue is selecting a reasonable number of the message boxes for our experimental study. After the above mentioned redundant folders were removed, 32 message boxes remained empty, as their owners did not create any additional folder. Then, we discovered that over 45% of other boxes still contained folders with less than 6 messages or with the total number of remaining emails 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 concept drift, we want to reduce the influence of other difficult aspects, like imbalance. This choice differs from the one already proposed in the literature (Bekkerman et al., 2004; Szopka, 2007), where only the largest email boxes were taken into account. We decided to finally choose seven boxes with

Table 1. Basic statistics of the selected email boxes.

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

approximately balanced multiple folders containing the possible high number of examples. The characteristics of chosen data sets are given in Table 5.1. As our boxes selection approach differs from the ones found in the literature, we cannot directly compare ours and the former results.

5.2. Constructing attribute representation

The next step consisted in transforming content of the email messages into an attribute–value representation suitable for learning classifiers. Although in this step we base on the whole dataset to select the most valuable attributes, the stream based nature of the algorithm presented in this paper remains true.

Each email was a text file written in English language including two major sections: header containing meta information and body text. Namely, the following fields of the header were used to describe examples:

Subject	the brief summary of the message topic;
From, Replay_to To, CC, Bcc	sender and recipient information parsed to get the complete email addresses or nicknames and were further treated as term-attributes;
Content-Type and Encoding	parameters describing the format of the message;
Date	time and date of email sending, transformed to its *UTC* value.

The *Date* element was not directly used as an attribute, but to chronologically order arriving data elements into the incremental sequence. No messages in Enron file contain any attachment files. Moreover, some parts of headers, such as server route, were not available. Generally speaking, the choice of the above elements of messages to create the representation of email is consistent with recommendations from earlier works, see e.g. Bekkerman et al. (2004), Klimt and Yang (2004), Stefanowski and Zienkowicz (2006), Szopka (2007).

In the case of header elements containing email addresses, we decided to apply the indexing technique known from the text document representation. Here, each term denotes a single email address. This is motivated by the fact that sender and recipient fields mostly contain a permutation sequence of distinct email addresses. Hence we tried to obtain the largest benefit from the additional meta data stored in the header section, which would be lost if sender and recipient fields were nominal attributes. The other content-type related fields of the header are considered as nominal attributes.

Note, that including header information into example description complies with results from Kmieciak (2009), where preliminary experiments were carried out with attribute space limited only to the body section. Results showed body section to be inefficient with respect to classification accuracy of J48 tree classifiers. Thus, information gain brought by the header, i.e. through subject and recipients, was significant. This observation is consistent also with experimental studies on other data sets (Stefanowski and Zienkowicz, 2006; Szopka, 2007).

Both subject field and the body of the message were processed as text documents. During *lexical analysis* (Baeza-Yates and Ribeiro-Neto, 1999) we chose terms in the *standard vector space model*, basing on typical tokenization and term elimination. The latest consisted in simple removal of terms which appeared only in one document, but also in English stopword list. Finally, term occurrences were transformed to attribute vectors.

As the number of obtained attributes was still very high, from 6007 to 11875, we decided to reduce this number. Following some inspirations from literature (e.g. Yang and Liu, 1999) and our earlier research on processing emails (Stefanowski and Zienkowicz, 2006; Szopka, 2007) we used feature selection method based on the Gain Ratio measure implemented in WEKA framework and reduced them to 3400 for each data set. This number resulted from technical reasons, considering available memory resources allowing us to sufficiently fast process data in the WEKA framework. Although this number was still high, we did not decrease it as in the further experiment we planned to use the decision tree algorithm $J4.8$, which can select the most important attributes to the tree. Let us also refer to the earlier study with some of the Enron data (Szopka, 2007), where other filtering methods, based on combination of Gain Ratio and χ^2 measures was used to get more compact (hundreds of features) representations.

All the data sets used for this study are publicly available in the form of *arff* files from WEKA framework and can be downloaded from `http://www.cs.put.poznan.pl/mkmieciak/enron` web page.

6. 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;

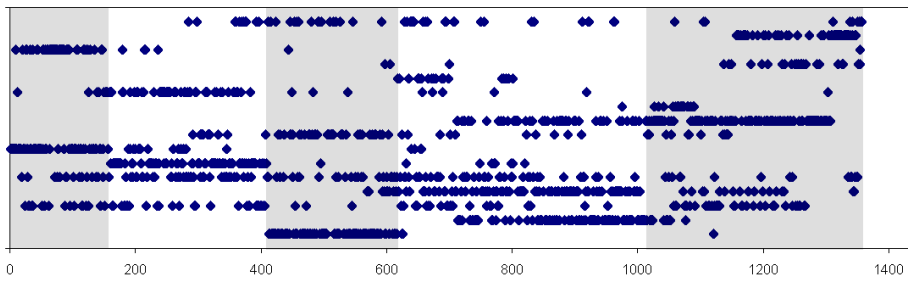

Figure 1. The time–space diagram of class distribution of the mann-k-mod dataset, with the message time stamp order increasing from left to right. Five manually created partitions are marked by shades of the background

2. Comparing our sem-supervised approach against two other popular fully supervised approaches.

Considering the first aim, we repeat our initial hypothesis that in the folder categorization we could expect sudden changes in the class distribution in the stream of incoming messages. It is reasonable to expect new class appearance when the user creates new folders, as well as when old class' members are no longer in the incoming email stream. To verify this hypothesis, we create the *time–spatial diagram* visualizing the distribution changes in time. Let us discuss the diagram of the mann-k-mod dataset (see Fig. 1). All the 16 classes (existing folders) in this set are marked vertically (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.

One can notice that changes of classes does not occur in the gradual fashion but suddenly. In the Fig. 1 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 is unchanged. This kind of *temporal locality* recognition seems to be important for the overall classification performance, as it enables drift resistant mining techniques. For instance, when data distribution is unchanged, a larger time horizon can be considered, providing more confident stream processing. In Fig. 1 we marked five partitions of the data stream, which may comply with the temporal locality of the data. In our opinion this clearly shows the presence of concept drift in the data, with sudden changes in the class distributions between the partitions. Time plots of the remaining data sets show similar characteristic, with at least four local partitions each. A more detailed presentation thereof is given in Kmieciak (2009).

The moments of the sudden concept drift found in Enron data directly influence classification of the new examples, as classifiers induced from the passed data may not able to correctly classify new data. Coming back to Fig. 1, where we see five local partitions representing five rather different sets of target

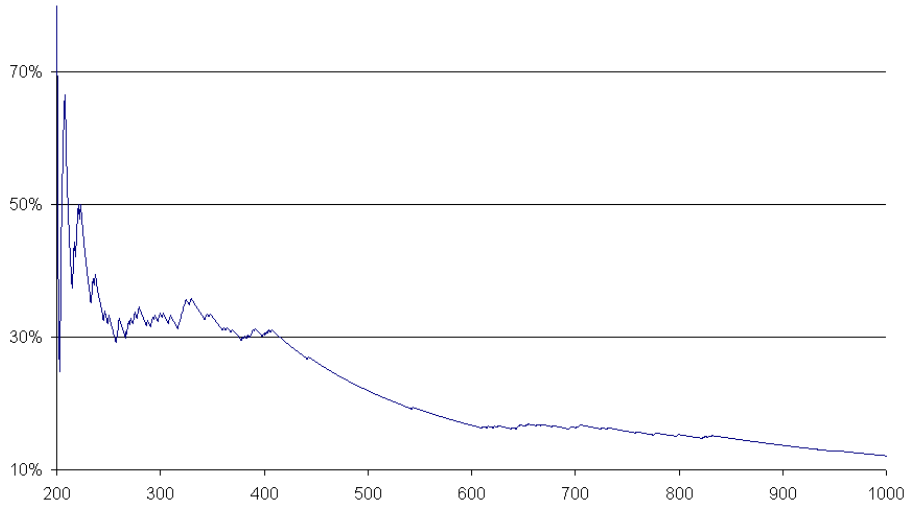

Figure 2. The cumulative classification accuracy plot for the hold out evaluation scheme applied to the mann-k-mod data set. The training set contains the $k = 200$ first examples. Each value of classification accuracy corresponds to the particular time stamp number of the test example

concepts, let us consider the case when the tree classifier is induced from the $training\_set\_size = 200$ first examples composing approximately the first partition. Fig. 2 shows 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 Fig. 1 we can say that the accuracy drop directly maps 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. Similar observations have been made for other data; for more details see Kmieciak (2009).

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 used in our approach. We compare plot shapes of the leaf statistic values and the cumulative accuracy measure. Fig. 3 shows relation between these values for mann-k-mod data set. Again, similar plots are observed for different moments of the data stream flow as well as for the remaining data sets.

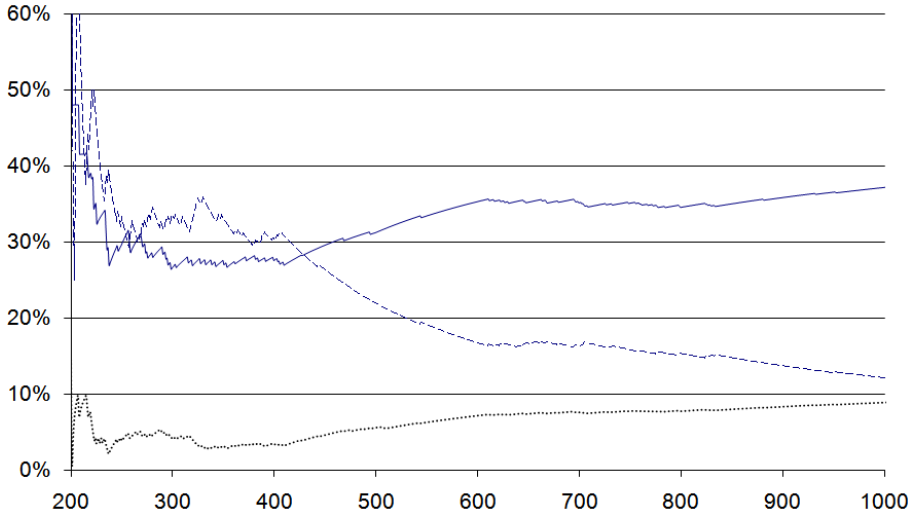

Figure 3. The relation between the cumulative accuracy measure (dashed line), leaf statistic value (solid line) and the expected error rate (dotted line) in the context of passing stream examples. The timestamps increase from left to right. Notice that the values of both indicators belong to $[0, 1]$

In Fig. 3, at the beginning, a certain number of examples from the processed stream is needed to stabilize the indicator 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), as it occurs around time stamp 400. The hyperbolic shape of this indicator seems to be complementary to the asymptotic cumulative accuracy decrease. For comparison, we also plot the behavior of the expected error (loss) indicator – originally considered in Huang (2008). 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.

The second aim of our experiment is to compare our semi-supervised approach against two different approaches to concept drift handling in the considered Enron data sets. We use our approach as described in Section 3, with two options for constructing the training sets. Both compared approaches required complete labeling of processed examples.

As the first approach, we chose a popular method based on *fixed periodic updates* of the classifier without direct detection of concept drift. Let us remark that due to frequent retraining and forgetting of older concepts this method could effectively adapt to possible changes. We chose it also due to its simplicity and popularity in the literature. 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 (Gama and Gaber, 2007).

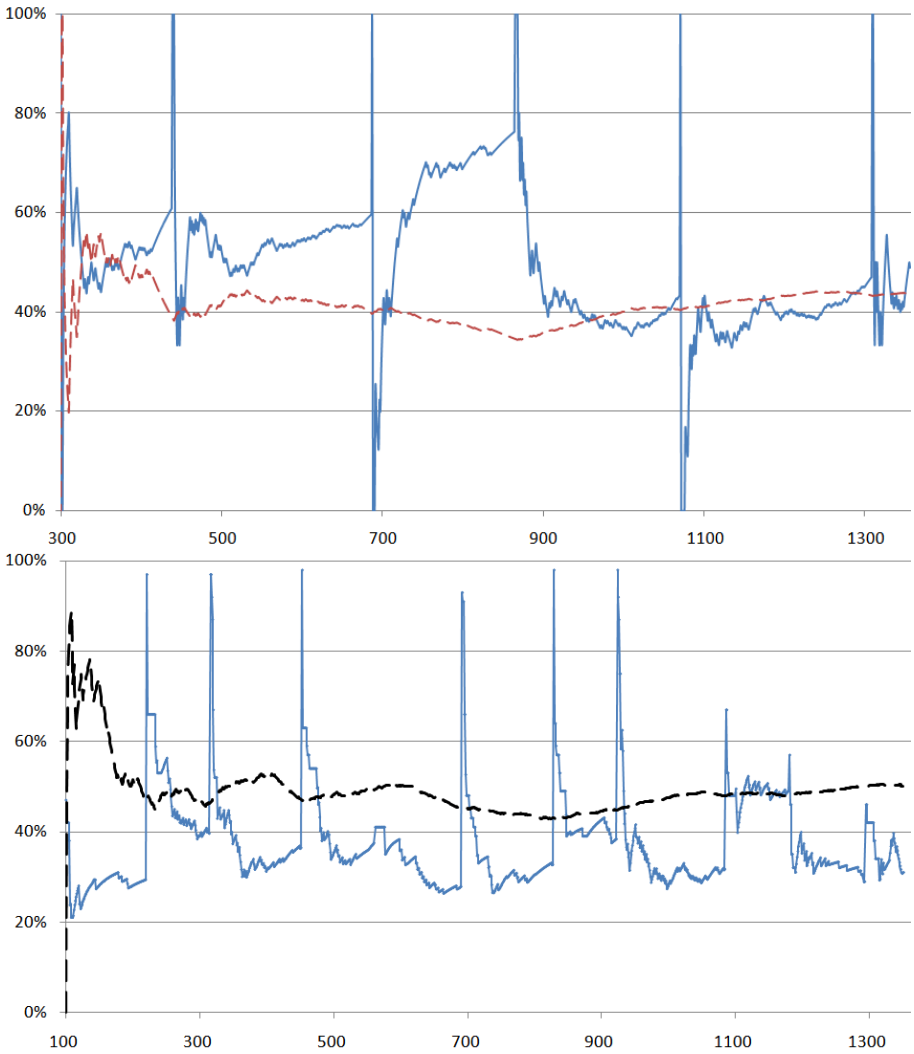

Figure 4. Comparison of drift detection in DDM (upper part of the figure) and our approach (lower part). For DDM the relation between the cumulative accuracy measure (dashed line) and DDM probability indicator (solid line) in the context of passing stream examples is shown. Similar relation is presented for our approach, with the solid line representing the change of leaf statistics. The timestamps increase from left to right. Again, the values of both indicators belong to $[0, 1]$

In the first option the training set is composed of all data between a specific time stamp called landmark and the present moment (Gama and Gaber, 2007). As we set landmark at the very beginning of the stream, the induced model incorporates both the very old and the latest data (it is definitely the most demanding from computational point of view and may not be realistic in case of much larger data than Enron). In the second option, the training set includes only *training_set_size* of the latest examples, thus limiting stream history and causing classifier to "forget" old concepts and focus on the new ones only.

However, unlike the incremental learning approaches to windowing where the classifier could be updated after reading a single example (see e.g. Widmer and Kubat, 1996), 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 block of examples needs to be labeled and then is added to earlier blocks while in the second option only the latest block of examples is used to retrain the classifier. In our experiments we decided to fix both the period length and the training set size.

The other chosen approach is the trigger based on *DDM* drift detection (Gama et al., 2004). We chose it as the best example of the trigger approaches. Since DDM requires complete labeling of examples, it potentially approximates errors more precisely than our detection method, processing unlabeled examples. In this context we could ask if our approach is able to detect concept drifts as well as the most popular equivalent supervised trigger approach

Sliding windows and our approach were implemented in Java basing mostly on the WEKA environment. Leaf statistics were observed using Quinlan's tree induction algorithm, $J4.8$ classifier. DDM was taken from the newest MOA project (Bifet and Kirkby, 2009)[6].

The evaluation scheme for all studied approaches is as follows. Once the tree classifier is induced from the chosen training set, it classifies the sequentially arriving data points. The *cumulative classification accuracy* measure is updated on each single data point, reaching the final value representing the 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. Moreover, the method remains consistent with opinions expressed in Bifet et al. (2009) saying that evaluation methods where the accuracy is incrementally updated and some examples are first used to test then to train are more appropriate for data streams than simple hold-out splits or cross-validation techniques, which do not take into account sequential appearance of examples, but are traditionally used in data mining. Besides the cumulative accuracy we analyze the number of retraining phases, as it directly refers to the expert's supervision for labeling examples.

[6] Available at `http://moa.cs.waikato.ac.nz`

The results of comparative experiments are given Tables 2, 3 and 4. The best classification accuracy is always marked in bold. For windowing approach, both versions achieve 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 frequent model update, determined by the period length (block, window size) parameter. Here, the value $training\_set\_size = 100$ examples seems to be the most adequate. It is somehow consistent with previously discovered frequent changes of classes in all data sets. On the other hand, frequent updates result in higher computational cost, as well as require more explicitly labeled examples. We will discuss this more precisely in the context of Table 5.

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 (the first number in range 0–100%), along with the number of retraining phases (the other number).

	Landmark window			Sliding window		
Training set length:	100	200	300	100	200	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 3. Evaluation results of DDM, for different sizes of the retraining set. The total accuracy values for each data set are shown, along with the number of retraining phases.

	Extending previous set			Limited horizon set		
Training set length:	100	200	300	100	200	300
farmer-d	65.3 12	**70.3** 7	66.5 5	60.6 11	58.2 7	62.6 10
germany-c	**51.2** 8	47.1 8	47.4 6	47.7 5	43.1 4	42.6 5
lokay-m	61.7 4	56.9 3	**64.9** 6	62.6 4	59.9 4	63.8 6
mann-k	38.6 8	18.0 3	**45.5** 7	34.3 8	20.9 6	23.3 4
mann-k-mod	23.1 5	31.0 6	31.6 5	36.3 7	12.7 3	**43.7** 6
rogers-b	64.3 5	59.4 5	56.9 5	**66.6** 5	54.7 5	65.2 5

Table 4. Evaluation results of our semi-supervised approach based on *drift detection* from unlabeled data, for different sizes of the retraining set. The total accuracy values for each data set are shown, along with the number of retraining phases.

	Extending previous set			Limited horizon set		
Training set length:	100	200	300	100	200	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 5. Evaluation results of Landmark/Sliding Windows, DDM and our semi-supervised detection-based methods. Best results are gathered.

	Accuracy (%)			Retrains number		
	L/S W	DDM	OurM	L/S W	DDM	OurM
farmer-d	**74.5**	70.3	64.5	23	7	**4**
germany-c	**55.3**	51.2	54.2	9	8	**6**
lokay-m	**67.6**	64.9	63.0	12	6	**5**
mann-k	**45.7**	45.5	43.3	15	**7**	13
mann-k-mod	50.3	43.7	**51.4**	13	**6**	11
rogers-b	**67.3**	66.6	66.2	8	5	**3**

The DDM trigger achieved slightly worse classification accuracy of 43.7% for mann-k-mod data, while L/SW and OurM methods reached 50.3% and 51.4%, respectively. However, it retrains the classifier significantly less times than sliding windowing and our method.

Considering results of our semi-supervised 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 those for windowing, we can say that depending on the data they are slightly worse only or comparable. Except for data sets farmer-d or lokay-m the differences are really small (up to 2%). Moreover, its classification performance is comparable to DDM.

However, we could stress that the number of retraining phases in our approach is much lower than for windowing and comparable to DDM (sometimes even smaller), what was the main motivation for our approach. For more information see Table 5. Looking mainly for comparison with DDM we can say that our approach can sufficiently well identify concept drift although it uses poorer information. Relations between change indicators in both approaches are shown in Fig. 4.

We calculated the number of labeled examples used by our detection method with the 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 from 64% to 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 instants and factors influencing it. 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. So, it is harder to observe trend slope changes of the distance measure as it is not clear enough and may be caused by 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–space diagrams we used. Moreover, the drift dynamics 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. 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 this data.

7. Conclusions

Our paper concerns constructing accurate classifiers that can adapt to concept drift in incrementally coming learning examples. The motivation for our research is that one cannot expect complete labeling for all incoming examples. Processing and evaluating classifiers on completely labeled examples is often assumed in the most well known approaches to handle concept drift; see e.g. the newest book by Gama (2010). We claim that it is more realistic to reduce demands for labeling to a relatively smaller part of examples in the stream, while the majority of processed examples are unlabeled. It also means that potential concept drift should be detected by analysing unlabeled data. Following such requirements we have presented a semi-supervised approach for handling sudden concept drift. It uses a decision tree classifier to monitor unlabeled data streams and to detect the concept drift basing on observing a trend of changes in probability distribution in the leaf statistics. Once it is identified, the classifier is retrained using a relatively small portion of the latest examples which need to be labeled.

Although we are inspired by some of existing methods, in particular active mining of data streams (Fan et al., 2004; Huang, 2008) we have considered different methods for analysing changes in probability distributions approximated by leaf statistics. Our contribution is also a different technique for detecting the concept drift as a result of discovering an increasing trend of differences between probability distributions in leaves (see more precise discussions in Section 3). Moreover, we focus our interest on detecting sudden concept drift, which has not been originally considered in Fan et al. (2004).

We have evaluated this approach in the experimental study with the folder categorization of Enron messages. Our next contribution is to clearly demonstrate that these Enron data sets can be considered as characterized by sudden concept drifts. Our analysis has also shown that changes of class definitions (appearance of new folders in the stream of emails) are quite frequent. As the number of real data sets for studying concept drifts is still limited (see Tsymbal, 2004), we present more details about pre-processing of the Enron data used and constructing their final representations, useful for applying typical learning algorithms (available in WEKA, MOA and other implementation platforms). We have to stress that our additional aim is to prepare such new "benchmarks" and make them available to research community[7].

In the next phase of the experiments with the Enron data sets we proved that sudden concept drift can be well identified by our approach. Looking at exemplary Figs. 2 and 3 one can notice that changes in our indicator have occurred at the expected moments in data streams. Moreover, our technique of identification of changes in leaf statistics has reacted better than the most related method based on loss functions (Fan et al., 2004). Other results clearly show that our method is comparable to the most popular trigger method for drift detection, DDM (Gama et al., 2004) – which assumes labeling of all processed examples (thus DDM uses much richer information than our method).

The final comparative experiment showed that our approach led to classification accuracy comparable to or only slightly worse than the popular windowing and DDM methods. The slightly better classification performance of the periodic learning of the classifiers with the shortest size of the data window may be better suited to quite frequent changes of classes in Enron data and other dynamic characteristics. However, windowing is definitely the most demanding from the computational cost point of view – see the number of retraining phases which is much higher than in DDM and our method (e.g. for *farmer-d* data windowing needs constructing new trees 23 times, while DDM needs 7 retrainings and our method slightly less). The most important experimental observation is that our approach significantly limits demand for labeling examples needed by the windowing method. In our opinion the cost of labeling is not negligible in most real life applications, so its reduction is worth a bit smaller classification performance.

[7]Data sets in format of *arff* are publicly available at the `http://www.cs.put.poznan.pl/mkmieciak/enron` web page

Concerning the computational aspect of the proposed approach, first of all, due to the smallest number of retraining / detecting possible concept drift and use of quite fast C4.5 algorithm this is a relatively fast approach. Let us also remind that we induce pruned trees which are quite small in our experiments. Using such small trees to classify new examples is also very fast. More difficult aspect concerns tuning parameters, in particular for detecting trend of our indicator (see Section 3). Our experiments have been carried out in controlled frameworks (as many similar studies in literature), so we could spend more time and efforts for repeating some experiments and checking several possibilities. In case of real huge data streams it is much more difficult. One should expect that some portion of examples (in particular in the starting phase) should be completely labeled and available for possible changes of default values of these parameters. It is the crucial requirement for our approach. The next parameter referring to the size of training examples or the size of retraining block is not so important, as we can generalize our approach to using on-line leaning of trees by specialized fast algorithms like VFDT (Very Fast Decision Trees, also known as Hoeffding Trees; Domingos and Hulten, 2000).

Besides giving up fixed size of the set of retraining examples, future research could concern a more active method of selecting the most informative examples for updating the classifier. Moreover, we could evaluate this approach on larger collection of data sets also characterized by different types of gradual drifts.

References

Baena-Garcia, M., Campo-Avila, J., Fidalgo, R., Bifet, A., Gavalda, R. and Morales-Bueno, R. (2006) Early drift detection method. In: *ECML PKDD 2006 Workshop on Knowledge Discovery from Data Streams*, 77-86. eprints.pascal-network.org/archive/00002509/01/EDDM.pdf

Baeza-Yates, R.A. and Ribeiro-Neto, B. (1999) *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston.

Bekkerman, R., McCallum, A. and Huang, G. (2004) Automatic categorization of email into folders: Benchmark experiments on Enron and SRI corpora. Technical Report IR-418, Center of Intelligent Information Retrieval, UMass Amherst.

Bifet, A. and Kirkby, R. (2009) Data stream mining: A practical approach. moa manual. Technical report, The University of Waikato.

Bifet, A., Kirkby, R., Holmes, G., Gavalda, R. and Pfahringer, B. (2009) New ensemble methods for evolving data streams. In: *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 139-148.

Bishop, C.M. (2007) *Pattern Recognition and Machine Leaning*. Springer, Berlin.

Brzezinski, D. and Stefanowski, J. (2011) Accuracy updated ensemble for data streams with concept drift. In: *Proceeding of the HAIS 2011 Conference.* **LNAI 6679**, Springer, 155–163.

Clark, J., Koprinska, I. and Poon, J. (2003) Linger – a smart personal assistant for e-mail classification. In: O. Kaynak et al., eds., *ICANN/ ICONIP 2003 Proc. of the 13th Intern. Conf. on Artificial Neural Networks.* **LNCS 2714**, Springer, 274–277.

Deckert, M. (2011) Batch weighted ensemble for mining data streams with concept drift. In: M. Kryszkiewicz et al., eds., *Proc. IMIS 2011.* **LNAI 6804**, Springer, 290–299.

Domingos, P. and Hulten, G. (2000) Mining high-speed data streams. In: *KDD '00: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, New York, 71–80.

Fan, W., Huang, Y., Wang, H. and Yu, P.S. (2004) Active mining of data streams. In: *Proceedings of the 2004 SIAM International Conference on Data Mining.* SIAM, 457–416, www.siam.org/ proceedings/datamining/ 2004/dm04.php

Gama, J. (2010) *Knowledge Discovery from Data Streams.* CRC Press, Boca Raton.

Gama, J. and Gaber, M.M. (2007) *Learning From Data Streams: Processing Techniques in Sensor Networks.* Springer, Berlin.

Gama, J., Medas, P., Castillo, G. and Rodrigues, P. (2004) Learning with drift detection. In: *SBIA Brazilian Symposium on Artificial Intelligence.* Springer Verlag, 286–295.

Han, J. and Kamber, M. (2006) *Data Mining: Concepts and Techniques.* Morgan Kaufmann, San Francisco.

Huang, S. (2008) An active learning method for mining time-changing data streams. In: *Proceedings of the 2008 Int. Symposium on Intelligent Information Technology Application, IITA'08.* IEEE Computer Society, Washington, 548–552.

Hulten, G., Spencer, L. and Domingos, P. (2001) Mining time-changing data streams. In: *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, New York, 97–106.

Kifer, D., Ben-David, S. and Gehrke, J. (2004) Detecting change in data streams. In: *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases.* VLDB Endowment, 180–191.

Klimt, B. and Yang, Y. (2004) The Enron corpus: A new dataset for email classification research. In: J.-F. Boulicaut et al., eds., *Proceedings of the ECML 2004 Conference.* **LNCS 3201**, Springer, 217–226.

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

KLOSGEN, W. and ZYTKOW, J.M. (2002) *Handbook of Data Mining and Knowledge Discovery.* Oxford Press, Oxford.

KMIECIAK, M.R. (2009) *Learning Multiple Classifiers from Text Streams.* Master's thesis, Poznan University of Technology, Poznań, Poland.

KUBAT, M. (1989) Floating approximation in time-varying knowledge bases. *Pattern Recognition Letters* **10** (4), 223–227.

KUNCHEVA, L.I. (2004) Classifier ensembles for changing environments. In: *Proc. of the 5th Workshop on Multiple Classifier Systems.* **LNCS 3077**, Springer, 1–15.

LEWIS, D.D. (1995) A sequential algorithm for training text classifiers: corrigendum and additional data. *SIGIR Forum* **29** (2), 13–19.

MASUD, M., GAO, J., KHAN, L., HAN, J. and THURAISINGHAM, B. (2008) A practical approach to classify evolving data streams: Training with limited amount of labeled data. In: *Proc. ICDM '08. Eighth IEEE International Conference on Data Mining.* IEEE Press, 929–934.

MASUD, M., GAO, J., KHAN, L., HAN, J. and THURAISINGHAM, B. (2009) Integrating novel class detection with classification for concept-drifting data streams. In: *Proc. ECML PKDD '09. Volume II.* Springer-Verlag, 79–94.

MITCHELL, T. (1997) *Machine Learning.* McGraw-Hill Education (ISE Editions), New York.

NISHIDA, K. (2008) *Learning and Detecting Concept Drift.* Ph.D. thesis, Graduate School of Information Science and Technology, Hokkaido University.

QUINLAN, J.R. (1993) *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Francisco.

SCHLIMMER, J. and GRANGER, R. (1986) Incremental learning from noisy data. *Machine Learning* **1** (3), 317–354.

SEBASTIANI, F. (2002) Machine learning in automated text categorization. *ACM Computing Surveys* **34** (1), 1–47.

STEFANOWSKI, J. and ZIENKOWICZ, M. (2006) Classification of Polish email messages: Experiments with various data representations. In: F. Esposito et al., eds., *Foundations of Intelligent Systems. 16th International Symposium ISMIS 2006. Proceedings.* **LNAI 4203**, Springer, 723–728.

SZOPKA, B. (2007) *Machine Learning and Text Processing Methods for Classification of Emails (in Polish).* Master's thesis, Poznan University of Technology, Poznan, Poland (supervisor J. Stefanowski).

TSYMBAL, A. (2004) The problem of concept drift: definitions and related works. Technical report, Dept. of Computer Science, Trinity College Dublin.

WIDMER, G. and KUBAT, M. (1996) Learning in the presence of concept drift and hidden contexts. *Machine Learning* **23**(1), 69–101.

WOOLAM, C., MASUD, M. and KHAN, L. (2009) Lacking labels in the stream: Classifying evolving stream data with few labels. In: *Proceedings of the ISMIS 2009 Conference.* Springer Verlag, 552–562.

Yang, Y. and Liu, X. (1999) A re-examination of text categorization methods. In: *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, 42–49.

Zliobaite, I. (2009) Learning under concept drift: an overview. Technical report, Faculty of Mathematics and Informatics, Vilnius University.