Journal of Universal Computer Science, vol. 19, no. 4 (2013), 462-483
submitted: 3/8/12, accepted: 20/2/13, appeared: 28/2/13 © J.UCS

Concept Drift Detection and Model Selection with Simulated Recurrence and Ensembles of Statistical Detectors

Piotr Sobolewski
(Wroclaw University of Technology, Poland
piotr.sobolewski@pwr.wroc.pl)

Michał Woźniak
(Wroclaw University of Technology, Poland
michal.wozniak@pwr.wroc.pl)

Abstract: The paper presents a concept drift detection method for unsupervised learning which takes into consideration the prior knowledge to select the most appropriate classification model. The prior knowledge carries information about the data distribution patterns that reflect different concepts, which may occur in the data stream. The presented method serves as a temporary solution for a classification system after a virtual concept drift and also provides additional information about the concept data distribution for adapting the classification model. Presented detector uses a developed method called simulated recurrence and detector ensembles based on statistical tests. Evaluation is performed on benchmark datasets.

Key Words: simulated recurrence, concept drift detection, detector ensembles
Category: H.2.8, I.2.6, I.5.2

1 Introduction

Classification systems are often not able to observe all the characteristics of data. E.g., due to lack of measurement instruments. As a result, some features in data stay hidden and are not taken into account when training the classification system and classifying new samples. In certain scenarios, some hidden features of data may affect a concept model in data stream, making the classification rules out of date. Such situation is called concept drift and it comes in many forms, depending on the type of change.

Change impetuosity categorizes the concept drift as gradual (slow changes, mild in nature) or sudden (abrupt changes, often referred to as "concept shift") [Narasimhamurthy and Kuncheva, 2007] and if the changes affect the data or class distribution, concept drift may be categorized as virtual or real respectively. Real concept drift is a change in class-conditional likelihoods when the prior distribution of input data patterns remains unchanged. Illustrative example of a real concept drift in a two class problem for a two dimensional feature space is presented in Fig. 1.

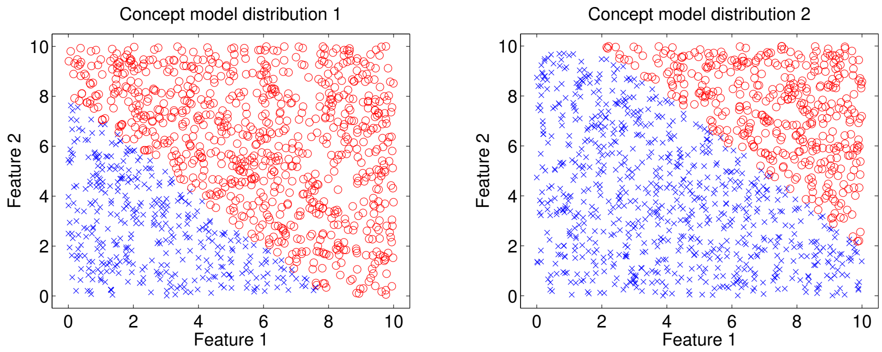

Figure 1: Real concept drift

On the other hand, a virtual concept drift is considered when the underlying data priors change [Yamauchi, 2010], as shown in Fig. 2. Such phenomenon can affect spam filtering applications, when the meaning does not change, only the data priors do (i.e., the relative frequency of the properties) [Wang et al., 2011].

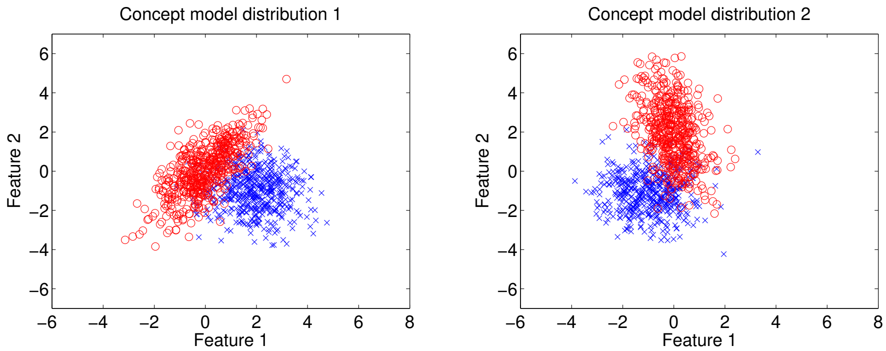

Figure 2: Virtual concept drift

In Fig. 3 an influence of a sudden virtual concept drift on the classification accuracy of an unprepared classification system is presented. The moment of concept drift is marked with a dotted red line. As it can be easily noticed, virtual concept drift may pose a serious threat for the performance of a classification system, which should be secured and minimized.

In general, approaches to cope with concept drift fit in one of the two categories [Greiner et al., 2002]:

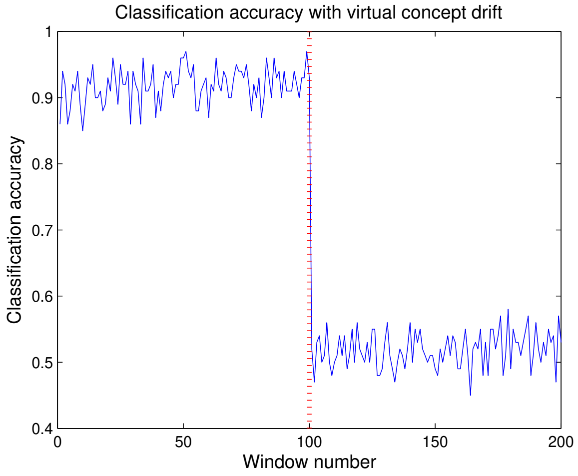

Figure 3: Classification accuracy with virtual concept drift (red dotted line symbolizes the moment of virtual concept drift)

- Approaches, which adapt a learner at regular intervals without considering whether the changes have really occurred [Ramamurthy and Bhatnagar, 2007]
- Approaches, which first detect concept changes and then adapt the learner to them [Lindstrom et al., 2011]

Adapting the learner is a part of an incremental learning approach [Muhlbaier et al., 2009]. Depending on the type of used learner, the model is either updated (e.g., neural networks [Cliff et al., 1992] or traditional decision trees [Gama and Medas, 2005]) or needs to be partially or completely rebuilt (as CVFDT algorithm [Hulten et al., 2001]). Ensebles of classifiers are also often evaluated [Smetek and Trawinski, 2011] as they are easy to scale and parallelize and they can adapt to change quickly by pruning underperforming parts of the ensemble [Attar et al., 2010].

In this article we focus on the methods, where detector and classifier are designed separately. Many detection algorithms base on a knowledge of object labels after the classification in order to detect concept drift, however as pointed out in [Zliobaite, 2010], such approach does not fit in the real scenarios. In general, concept drift detection algorithms can be divided into three types, depending on the assumption about the amount of costly knowledge regarding the true class labels available for the algorithm, namely:

- Supervised algorithms – assuming access to classification performance measures or true class labels, detecting concept drift on the basis of classifier's

accuracy or analysis of class likelihoods [Kifer et al., 2004]
- Semi-supervised algorithms – assuming limited access to classification performance measures or true class labels, also detecting concept drift on the basis of the properties of data when such knowledge is not available, e.g. active learning [Kurlej and Wozniak, 2012] [Greiner et al., 2002]
- Non-supervised algorithms – assuming no access to classification performance measures or true class labels and basing only on the properties of data, detecting concept drift on the basis of attribute value distribution, cluster memberships or classifier's support levels [Lane and Brodley, 1999] [Spinosa et al., 2008]

In this article, we explore the possibilities of detecting concept drift in data streams without any supervision. It is worth noting, that such approach has some limitations:
- firstly, it is suited only for virtual concept drift, as the real concept drift is undetectable by analyzing solely the properties of data,
- secondly, there are certain situations when also virtual concept drift is impossible to detect, e.g. when classes swap places, as presented in Fig. 4.

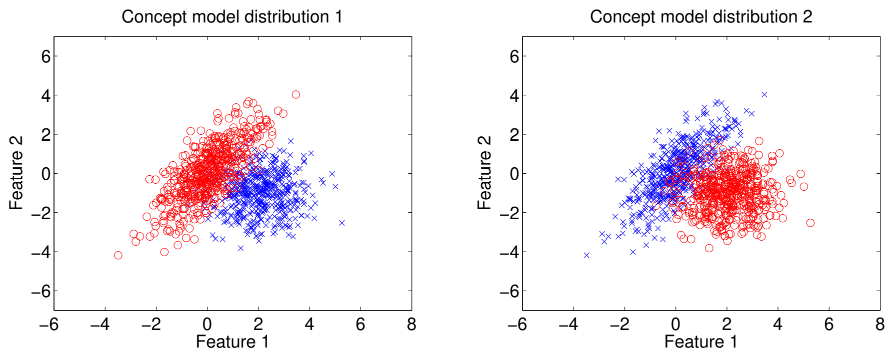

Figure 4: Special case of virtual concept drift

The main contribution of the article is an unsupervised concept drift detection method based on prior knowledge about the possible data distributions. Problem of concept drift detection is extended to concept model selection and the drop of classification accuracy is minimized by using temporary classification models. Additional information regarding the data distribution can also improve the process of the model adaptation.

The remainder of the article is organized as follows. Section 2 is an overview of unsupervised concept drift detection algorithms, describing the methods used in the article. Section 3 presents a method for creating temporary concept models on the basis of prior knowledge. Next, Section 4 covers the evaluation of the method with a discussion on the results and the article concludes in Section 5.

2 Detecting virtual concept drift without supervision

Unsupervised detection of virtual concept drift is most often performed with statistical tests [Markou and Singh, 2003], which check whether a data chunk (a group of consecutive records in data stream, also referred as data window) comes from the same distribution as the reference data. Obviously, not all statistical tests are suited for this task, e.g. two-sample parametric tests such as a T2 statistic [Hotelling, 1931] assume a specific distribution, which might not be a correct approach in the real data case, as the samples may include records from several classes, each described by a different distribution. Also, the distributions may not be similar to any standard distribution, what moreover suggest non-parametric tests for the task of unsupervised concept drift detection. Examples of such tests include:

- CNF test [Dries and Rückert, 2009]
  Approach introduced in [Dries and Rückert, 2009], describes the data by vectors of binary features, assigned by discretizing attributes into sets of bins. It then creates a set of Boolean attributes A, which "covers" all of the examples in the reference set of data X, meaning that each "true" feature in set A is the same as in at least one of the vectors describing the data points in X. Next, another set of data $\overline{X}$ is drawn from the same distribution as the data in X, represented as binary vectors, and compared to set A, by calculating parameter c_i for each example x_i in $\overline{X}$, which is measured by counting the number of clauses in set A, which do not "cover" x_i. When a data window DW is tested to check if it comes from the same distribution as X, a sequence of parameters c_i is measured for all data samples in the window and compared with the sequence of c_i's obtained by comparing the distributions of data in X and $\overline{X}$ by applying a Matt-Whitney test. If the difference is insignificant, all data is considered to come from the same distribution, otherwise a difference in distributions is detected.
- The Wald-Wolfowitz Test [Friedman and Rafsky, 1979]
  The multivariate version of the Wald-Wolfowitz test [Friedman and Rafsky, 1979] constructs a complete graph, with examples as vertices and distances between them as edges. Graph is then transformed into a forest and a test statistic is computed basing on the amount of trees.

Also, non-parametric univariate statistical tests are often used for detecting concept drift in data distribution [Sheskin, 2011]:

- Two-sample Kolmogorov-Smirnov test,

The two-sample Kolmogorov-Smirnov test [Smirnov, 1948] is non-parametric, as it makes no assumption about the distribution of data and therefore can be deployed on any data.
For the two-sample test, a Kolmogorov-Smirnov statistic is computed as

$$D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)| \tag{1}$$

where $F_{1,n}$ and $F_{2,m}$ are the empirical distribution functions of samples computed as:

$$F_n(t) = \frac{1}{n}\sum_{i=1}^{n} 1\{x_i \leq t\}, \tag{2}$$

where $(x_1, ..., x_n)$ are independent and identically distributed (i.i.d.) random variables laying in the real numbers domain with a common cumulative distribution function. The statistic is used to perform a KS-test to reject the null hypothesis at level α by computing:

$$\sqrt{\frac{nm}{n_m}} D_{n,m} > K_\alpha, \tag{3}$$

where K_α calculated from:

$$Pr(K \leq K_\alpha) = 1 - \alpha, \tag{4}$$

and K is a Kolmogorov distribution computed as:

$$K = \sup_{t\in[0,1]} |B(t)|, \tag{5}$$

$B(t)$ being the Brownian bridge [Revuz and Yor, 2004].
In short, the Kolmogorov-Smirnov test compares the distributions of two samples by measuring a distance between the empirical distribution functions, taking into account both their location and shape.

– Wilcoxon rank sum test,
Wilcoxon rank sum (also called Mann–Whitney–Wilcoxon) test [Wilcoxon, 1945] is a non-parametric alternative to the two-sample t-test, based solely on the order in which the observations from the two samples fall.
The test assumes, that all observations are independent from each other and can be ordered by their value, therefore if the test is performed on the data which are categorical, it has to be mapped to the numerical values.
The test ranks all observations regardless of which sample they are in by ordering them from the greatest to the lowest value. Then, a statistic is computed for each of the samples as:

$$U = R - \frac{n(n+1)}{2}, \tag{6}$$

where n is the sample size and R is the sum of the ranks in this sample. In order to reject the null hypothesis that both samples come from the same population, a lower value of U from both samples is chosen and consulted with the significance tables.

– Two-sample T-test.
  Two-sample t-test is one of the most popular tests used in economics and quality measures. It calculates the t-statistic on the basis of the means $\overline{x_1}$, $\overline{x_2}$, standard deviations s_1, s_2 and the numbers of observations n_1, n_2 in each sample by:

$$t = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}, \tag{7}$$

  which is then compared to the critical $t - value$ taken from the significance tables, with regard to the number of the degrees of freedom, k:

$$k = \{ n_1 - 1, \text{if } n_1 < n_2 n_2 - 1, \text{if } n_1 > n_2 n_1 + n_2 - 2, \text{if } n_1 = n_2 \tag{8}$$

  The true outcome of the test means the rejection of the null hypothesis that both samples originate from the same distribution.

There are many more algorithms for detecting concept drift in data streams described in the literature, however vast majority of them bases on the knowledge of class labels available after the classification [Alpaydin, 2010], what is against the assumption of non-supervision. In this article, we are going to focus on the methods which detect concept drift on the basis of the properties of data only, using five mentioned above detection algorithms combined in an ensemble.

3 Simulated recurrence

Approach assumes that, although we should not have access to the expert's knowledge during the operation of the classification system, we still may have some information regarding the nature of the problem available beforehand. Namely, we assume that an individual interested in deploying a classification system for classifying a stream of data, possess the knowledge about possible concepts of distribution models which may arise in the data. A real life example of such situation would be an owner of a grocery store, who knows generally what types of data may be observed and therefore may provide the knowledge about possible distributions among the classes of data e.g., one product may become more popular (due to an aggressive marketing campaign or a new trend in fashion), what may affect the demand on the other products. It is worth noting, that usually every change has its own limitation, as a system for weather prediction would not have to consider a glacier in July or floods in the middle of desert.

In this paper we are exploiting the opportunity created by the knowledge about possible data and class priors. The method is called simulated recurrence and it was first introduced as an extension to classification algorithms coping with recurring concept drift in [Sobolewski and Woźniak, 2011]. Recurring concept drift facilitates an assumption, that concepts may recur in the data stream, namely after the concept shifts, the data may follow the same model as previously observed. When it happens, a classification system can use the gathered knowledge regarding this model and quickly adjust the active classification rules. The aim of simulated recurrence approach is to re-create such situation in a non-recurring concept drift scenario, where recurring concepts are replaced with artificially simulated concepts. The concepts are simulated on the basis of available knowledge about the models.

The knowledge is represented by the data samples, on basis of which arificial datasets are generated, representing different concepts. A method for simulating the data distribution is described in Section 3.1.

On the basis of each simulated concept an artificial classification model is created. Although an artificial model is not a perfect classification tool, it may serve as a temporary solution for classifying samples following a concept model similiar to the simulated one. The samples in the stream may also be used for updating the artificial classification model or creating the new model with an unsupervised learning approach. The method for selecting a concept model is described in Secion 3.2.

The simulated recurrence [Sobolewski and Woźniak, 2012] is mainly used to improve the overall efficiency of the classification system coping with concept drift, solely by decreasing the error-rate. The approach has obtained promising results, improving the performance of classification systems by around 25%. The main contribution of this paper compared to our last work is the use of simulated recurrence, namely we introduce it as a tool for concept drift detection and model selection, rather than an extension for classification module. It does not affect the classification system at all, working independently as an enhanced detector.

A typical concept drift detection system aims to determine whether the data comes from the same distribution as the reference data by analyzing the p-values returned by the statistical tests. If the null hypothesis is rejected, the data is considered to arise from a different distribution and a signal about possible concept drift is passed to the classification system. Otherwise, the data is considered to follow the distribution already known and the classification system is not alarmed about a danger of decreasing accuracy. Such binary detection does not carry much valuable information. Simulated recurrence extends the functionality of a binary detector by including the information about the possible data distribution in the new concept, which may be used for enhancing the adaptation of the classification model, responding more quickly to the change in concept.

During the process of classification model's adaptation, a temporary artificial classifier may be used to minimize the drop of the system's performance.

Classification system's mode of operation with simulated recurrence is described in Fig. 5 as a pseudo-code.

3.1 Simulating a data distribution

Simulated recurrence assumes, that before the system begins the classification routine, partial knowledge regarding the possible data distribution patterns encounterable in the data stream is available. For the purpose of experiments we assume that this knowledge consists of the samples which reside in the border and in the centers of the the class clusters. An example of these points is presented in Fig. 7, with regard to the true distributions presented in Fig. 6. Basing on these data points, class covariance matrices and the means are calculated and remaining samples are generated following a Gaussian distribution. The method for selecting data points on real data is explained in details in Section 4.1.3.

3.2 Selecting the concept model

As simulated recurrence replaces the detector with a model selector, we propose a method for selecting the concept models based on the properties of data samples in the data window as described in Fig. 8.

In the first step, each of the non-parametric statistical tests described in Section 2 performs a check for every concept dataset whether the data in a window comes from the same distribution as the corresponding concept data. The checks are performed for the reference dataset and simulated data, and each check results in a *p-value* representing the probability that both distributions are the same. In the case of univariate statistical tests, one test is performed for each single feature and the summary result for one concept is a sum of *p-values* obtained for all features.

```
Notations:
  DW - data window,
  CM_i - the i-th concept model,
  CL_i - the i-th classification model,
Algorithm:
  Draw new DW from data stream,
  Select CM_i closest to DW,
  Classify samples in DW using CL_i.
```

Figure 5: Pseudo-code of classification with simulated recurrence

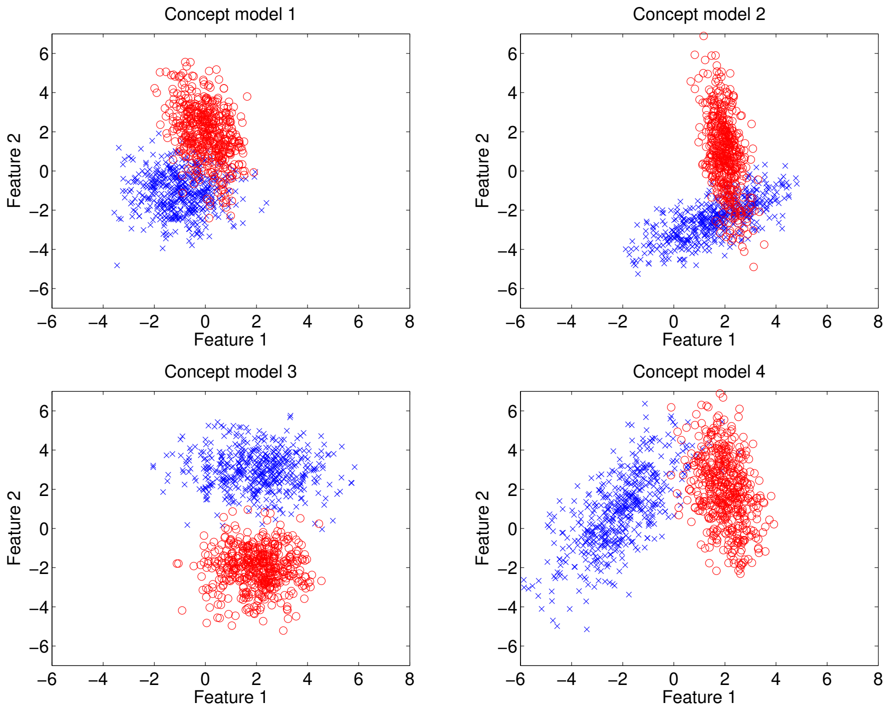

Figure 6: An example of all possible concepts (data distributions) in a classification scenario.

Next, the concept distribution which is the closest to the distribution of data in a window according to one statistical test (namely, concept distribution which has the lowest *p-value*), receives one vote from the corresponding test statistic.

Last step is a selection of the model which has the highest number of votes. In the case when more than one concept model has the most votes, the selection from performed randomly from the top candidates.

4 Experimental evaluation of algorithm

The aim of experiments is to evaluate the ability of the simulated recurrence method to accurately select the classification models when concept drift occurs. As a second step, the mean classification accuracy after concept drift is also evaluated in order to estimate an influence of the simulated recurrence approach on an unprepared classification system.

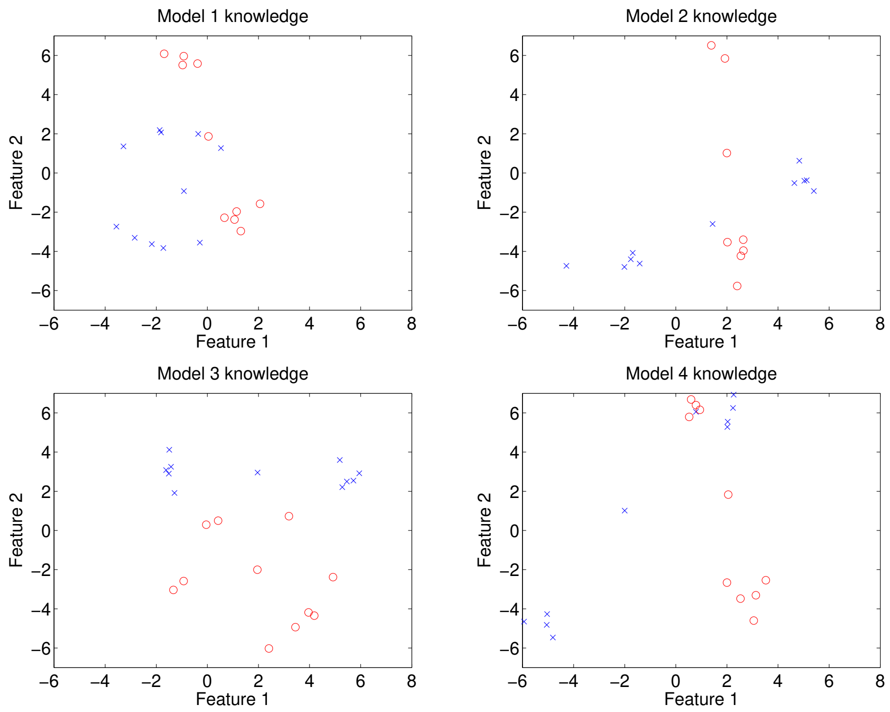

Figure 7: Information provided to the classification system a priori, describing the possible concept in Fig. 6.

4.1 Setup of the experiments

4.1.1 Measures

There are no universal measures for evaluation of concept drift detection algorithms. The most intuitive metrics to estimate the efficiency of the proposed method are:

- Concept model selection accuracy (sensitivity / specificity)
- Mean classification accuracy

The first measure, the model selection accuracy presents the ability of the algorithm to accurately identify from which concept distribution the data window origins.

The second one is the most often used metric for estimating the classification algorithm's performance, however in this case it is used mainly to show how the temporary artificial classifiers manage to reduce drop of classification accuracy compared to the unsecured classification system. In the meantime, the system may adapt to the new concept model, what is not a subject of this article.

```
Notations:
  DW - data window,
  ST_i - the i-th test statistic,
  CM_i - the i-th concept model,
Algorithm:
  Draw new DW from data stream,
  FOR i = 1 : n, n - number of test statistics
    FOR j = 1 : m, m - number of concept models
      Calculate p-value_j as a result of test that DW comes
       from the same distribution as CM_j,
    ENDFOR
    Give vote for CM, which has the lowest p-value,
  ENDFOR
  IF more than one CM has the most votes,
    Select randomly CM from the ones having the most votes.
  ELSE,
    Select CM with the most votes.
  ENDIF
```

Figure 8: Pseudo-code of model selection procedure

By combining both measures, it can be inferred e.g., which concept is the most critical, difficult to detect and resulting in severe drops in classification accuracy and which can be inored, producing minor drops in the system's performance.

4.1.2 Classifier

As a classification model, we use a standard Parzen classifier [Duda et al., 2001], which bases on estimating probability density for each class using a non-parametric approach. When computing output for a new observation, the contribution of each training example is integrated. The contribution is modeled by a kernel function (in our example, we use Laplace kernel as it is computationally less demanding than a popular Gaussian kernel) and is influenced by the smoothing parameter (kernel width). The smoothing parameter is optimized using EM algorithm optimizing cross-validated log-likelihood.

4.1.3 Datasets

Algorithm is evaluated on the benchmark datasets from UCI machine Learning Repository [A. Asuncion, 2007]. These datasets contain samples which are following a static distribution, what implies the need for simulating the changes in concept model in order to evaluate a method for classification of data affected by concept drift. There are several methods described in the literature which solve this issue, e.g.:

- Switching features [Zliobaite and Kuncheva, 2009] - the method based on switching the feature values for sets of data samples while keeping the class labels, it simulates a virtual concept drift
- Rotating values [Wang et al., 2005, Ramamurthy and Bhatnagar, 2007] - the method based on changing the values of certain features in a dataset for one class with another, it simulates a real concept drift
- Treating class data as a single concept [Vreeken et al., 2007, Dries and Rückert, 2009] - the method based on picking two most populated classes from the dataset and treating them as a single class, but in a different concept, it simulates virtual concept drift
- Rotating data around the center of the feature space [Zliobaite, 2008]
- Simulating environment changes in a 3-D driving game [Lindstrom et al., 2008]

Choice of the method for simulating concept drift is determined by the type of concept drift aimed for evaluation in the classification scenario. Most of the methods use parameters to control the way in which concept drift affects the data. These parameters are often used to tune the concept drift to the classification algorithm, making the results less credible. Also, many methods simulate a real concept drift in data, not suited for an unsupervised system. The most fair and independent way of simulating virtual concept drift described in the literature seems to be a method pioneered in [Vreeken et al., 2007] and later used in [Dries and Rückert, 2009], which takes data samples from the two most populated classes and treats them as data from different concept models. Such approach limits the classification problem to a single class and only two concepts, therefore we extend this method to two classes per concept and number of possible concept models to the number of classes in the dataset divided by two.

First, the data is ordered descending by class population. Next, if the number of classes in dataset is odd, the least populated class is removed. Lastly, the classes are paired and divided into concept datasets, each representing a two-class problem. This process has been shown in Table 1 on the example of the "Page-blocks" UCI dataset.

Such concept simulation method determines the choice of UCI datasets, as the original datasets need to have at least 4 classes in order to simulate at least two concept model distributions. Concept 1 is a reference model used for preliminary classifier training. The datasets chosen for experiments with the number of features and number of samples in each class ares presented in Table 2.

Table 1: Page-blocks dataset divided into concept datasets

	Original Class	Num. samples	Concept Class	
Original dataset	text	4913	Class 1	**Concept set 1**
	horiz. line	329	Class 2	
	picture	115	Class 1	**Concept set 2**
	vert. line	88	Class 2	
	graphic	28	removed	-

Table 2: Description of UCI datasets selected for experiments. *(Cl.x denotes number of elements from the x-th class)*

Dataset	Num. of feat.	Number of samples in classes									
		Cl.1	Cl.2	Cl.3	Cl.4	Cl.5	Cl.6	Cl.7	Cl.8	Cl.9	Cl.10
car	6	1210	384	69	65	0	0	0	0	0	0
heart-c	13	160	54	35	35	13	0	0	0	0	0
mfeat-mor	6	200	200	200	200	200	200	200	200	200	200
nursery	8	4320	4266	4044	328	2	0	0	0	0	0
optdigits	64	389	389	387	387	382	380	380	377	376	376
page-blocks	10	4913	329	115	88	28	0	0	0	0	0
pendigits	16	780	780	780	779	778	720	720	719	719	719
vehicle	18	218	217	212	199	0	0	0	0	0	0
yeast	8	463	429	244	163	51	44	35	30	20	5

In order to represent the knowledge available beforehand to create a simulated recurrence, we assume that 10% of border points and a central tendency of each class are available. In order to estimate a central tendency for a single class of data, a following equation is deployed:

$$\overline{\mathbf{x}}^{(k,j)} = \frac{1}{N} \sum_{i=0}^{N} \mathbf{x}_i^{(k,j)} \ , \tag{9}$$

$\overline{\mathbf{x}}^{(k,j)}$ is a sample mean vector of all N real observations belonging to class j in the k-th reference concept dataset,

$\mathbf{x}_i^{(k,j)}$ is the i-th observation belonging to class j in the k-th reference concept dataset.

Next, the 10% border points for a class j are chosen according to the following process:

First, find a point which is the most distant from the other points:

$$\mathbf{p}_1^{(k,j)} = \arg\max_{\mathbf{X}^{(k,j)}} \sum_{i=0}^{N} d(\mathbf{x}^{(k,j)}, \mathbf{x}_i^{(k,j)}) \ , \tag{10}$$

where $d(\cdot,\cdot)$ denotes *Euclidean distance.* Afterwards, the remaining $\frac{N}{10} - 1$ points are selected iteratively as the points which are the most distant from the last selected point and the remaining points.

Covariance matrix $\mathbf{Q}$ for class j in the k-th reference concept dataset is estimated according to the equation:

$$\mathbf{Q}^{(k,j)} = \frac{1}{3(N-1)} \sum_{i=0}^{N} (\mathbf{x}_i^{(k,j)} - \overline{\mathbf{x}}^{(k,j)})(\mathbf{x}_i^{(k,j)} - \overline{\mathbf{x}}^{(k,j)})^T \ , \tag{11}$$

where $\overline{\mathbf{x}}^{(j)}$ is a class sample mean vector calculated according to equation (1).

Having the covariance matrix and a mean, the samples in the k-th concept dataset's class j are generated according to a standard Gaussian distribution.

Number of generated samples is the same as the number of samples of the corresponding class in the reference dataset.

Experiment plan.

Experiments are divided into test runs, which are performed independently on randomly drawn samples, grouped into windows of data. A single test run consists of performing the Model Selection procedure (Fig. 8) on one window of data from a single concept dataset and classifying the data window with the corresponding classification model (either reference or temporary, depending on the concept model selection).

For a single scenario, 100 test runs are performed for each concept model. The size of data window is constant and set to 15 samples. Model selection accuracy is measured together with an average classification accuracy on the data window. Results are gathered in two tables and each scenario is represented by a separate row.

4.2 Results

Before calculating a mean accuracy, all test runs are compared and statistically validated with a 5% significance level of rejecting the null hypothesis with a paired t-test [Rubin, 1973]. Significantly better results are marked with a bold font. Table 3 summarizes the specificity and sensitivity of the Model Selection procedure. Table 4 compares the accuracy obtained by a system using a single

Table 3: Concept model selection accuracy [%]

Dataset	Specificity	Sensitivity			
	Reference concept	Concept model 1	Concept model 2	Concept model 3	Concept model 4
car	100	34	-	-	-
heart-c	100	56	-	-	-
mfeat-mor	98	86	78	88	76
nursery	99	53	-	-	-
optdigits	100	0	0	0	0
page-blocks	100	58	-	-	-
pendigits	100	99	95	98	96
vehicle	99	75	0	0	0
yeast	100	37	83	69	86

classification model trained on the reference concept dataset with the accuracy of the system equipped with the concept drift detection module based on the simulated recurrence.

4.3 Discussion of the results

Results of classification accuracy presented in Table 4 depend on several factors:
- Concept model selection accuracy
- Concept model representation
- Classification algorithm
- Difficulty of the classification scenario

The concept model selection accuracy depends on the degree to which the concepts differ. Selected concept models might not represent directly the real concepts, however if the data distribution patterns are similar, the drop in the classification accuracy should not be major. In Table 4, the classification accuracy of the system without the model selection algorithm (only the base classifier trained on the reference concept data), the proposed method and a system equipped with a perfect selector is compared. The latter is a scenario where model selection is always correct (100% accuracy) and it represents the degree to which the classification models trained on the simulated concept datasets are able to classify the corresponding concept data. The scores obtained by the perfect selector can be optimized either by improving the classification algorithm or designing a better method for representing the real concept basing on the provided knowledge a priori. Comparison of the three values for each concept

Table 4: Classification accuracy [%]

Dataset	Reference concept			Concept model 1			Concept model 2			Concept model 3			Concept model 4		
	no sr	with sr	*pr sel*	no sr	with sr	*pr sel*	no sr	with sr	*pr sel*	no sr	with sr	*pr sel*	no sr	with sr	*pr sel*
car	100	100	*100*	41	**57**	*93*	-	-	-	-	-	-	-	-	-
heart-c	78	78	*78*	48	53	*54*	-	-	-	-	-	-	-	-	-
mfeat-mor	97	96	*97*	59	59	*65*	18	**76**	*93*	48	**84**	*84*	91	90	*91*
nursery	100	100	*100*	32	**67**	*97*	-	-	-	-	-	-	-	-	-
optdigits	100	100	*100*	100	100	*100*	36	36	*100*	39	39	*100*	43	43	*100*
page-blocks	94	94	*94*	56	56	*56*	-	-	-	-	-	-	-	-	-
pendigits	100	100	*100*	90	**99**	*99*	30	97	*100*	93	**100**	*100*	51	**96**	*99*
vehicle	100	100	*100*	33	**73**	*85*	-	-	-	-	-	-	-	-	-
yeast	64	64	*66*	64	**74**	*92*	48	78	*85*	58	**77**	*82*	96	98	*100*

shows how the presented method protects the unprepared classification system from the effects of the virtual concept drift and also how much better it could perform if the model selection procedure was more accurate.

Standard Gaussian distribution is used to represent the corresponding data distribution in the benchamrk datasets, which is a very simple method that leaves a broad field for optimization. On the other hand, the classification accuracy scores obtained by the perfect selector show, that the classification models trained on the simulated data achieve almost perfect results in most of the scenarios.

The classification algorithm is not within the scope of discussion in this article, however a choice of a more proper method for each scenario could result in higher accuracy scores.

The last factor is the difficulty of the scenario, which depends on the data used. As benchmark data used in experiments does not condsider concept drift, it has to be artificially implemented. A different method of applying concept drift in bechamrk data would also result in different scores.

The results presented in Table 3 show how the presented method is able to identify the concept model distribution, which describes the samples in the data window. These values depend on:

– Model selection algorithm
– Real distribution representation

The model selection algorithm described in Section 3.2 is based on an ensemble of statistical detectors. This approach leaves a field for optimization and further research, as there are many other concept drift detection methods described in literature, which could be also used for this purpose, such as [Kuncheva, 2011, Zhou et al., 2009].

Representing the real concept distribution by analyzing the prior knowledge is also an open subject for discussion. First of all, knowledge available for the system beforehand is not strictly defined and it does not leave much options for the concept model simulation. The standard Gaussian distribution generalizes the most true distributions and therefore is the most universal choice. Practical application of the method would definitely require a more specific tuning.

The concept model selection accuracy is the most important in the cases when reference concept and the new concept are very diverse. The degree of the similarity of concept models may be represented by the drop of classification accuracy in the system equipped only with a reference classifier. In several scenarios the difference between the concepts is significant, making the model selection meaningful, e.g. for *mfeat-mor* concept 2 and concept 3 or *pendigits* concept 2 and concept 4. On the other hand, in some scenarios the classification accuracy for the reference concept data and the new concept data almost does not differ e.g., for *mfeat-mor* concept 4, *optdigits* concept 1, *pendigits* concept 1 and concept 3 or *yeast* concept 1 and concept 3, suggesting a low diversity of the mentioned concept models.

The scores can be analyzed more specifically by taking into consideration the characteristics of each experimental scenario.

Scenarios, which benefit the most from the proposed method are the ones, which experience the most severe accuracy drops by the reference classification system and also achieve a good model selection accuracy. Examples are *mfeat-mor* and *pendigits*, which achieve very high model selection accuracy scores for each concept, *vehicle* where the drop of classification accuracy is significant and *yeast*, which is a special case because the reference classifier achieves significantly better results for concept 4 data than for the reference data. Such situation is caused mainly by the low number of samples available for each class in the original dataset, what influences the ability of training the classification models.

Low model selection accuracy for *car* and *nursery* datasets result in a slight accuracy increase compared to the reference model. A significant difference between the classification accuracy achieved by the simulated recurrence approach and the perfect selector shows a major field for improvement and suggests, that for this scenario a better model selection algorithm should be designed.

Low model selection accuracy can also be observed for *heart-c*, *page-blocks* and concept 2 of *yeast* scenarios, while not each of these cases is critical. Each of these scenarios has very imbalanced number of samples in classes. For *heart-c* and *page-blocks*, the perfect selector did not improve the classification accuracy of the system, therefore the influence of applied virtual concept drift has been minimized.

Better model selection scores have been achieved for the scenarios, where number of classes is balanced, namely *mfeat-mor*, *pendigits* and *vehicle*, although all these scenarios are characterized by the highest number of possible concepts, what should rather suggest worse scores (higher number of possible selections means higher probability of a mistake). Surprisingly, the results are opposite. This suggests, that ratio of samples in each class has a high influence on the concept model selection algorithm's performance. An exception is *optdigits*, which is an interesting dataset, as it is characterized by a large number of features compared to the other experimental scenarios. It is also the only scenario for which the model selection algorithm fails completely every time for each of the concept models. These facts suggest, that the model selection method presented in the article is not suited for the highly dimensional data. This phenomenon needs to be analyzed more thoroughly and tested on different highly dimensional scenarios, as the results presented by the perfect selector show, that the classification accuracy drop can be minimized almost entirely with the same classification algorithm and the same method for representing the concept models.

5 Conclusions and Future Works

Proposed method enhances the detection of concept drift by providing additional information regarding the possible concept model distribution in the data without supervision. Available prior knowledge regarding the possible data distributions is used to create temporary classification models, which decrease the drop of classification accuracy when the data distribution is affected by a virtual concept drift. This knowledge is also used for simulating the concept data, what allows the use of statistical tests for concept model selection. In this paper a majority voting ensemble approach is evaluated in order to minimize the influence of very sensitive test statistics and to improve the overall quality of the insensitive detection algorithms.

The classification models are used based on model selected by the detector ensemble. Trained beforehand on the simulated concept data, they are not updated during the operation of the classification system. Adaptation of the models is still an open issue, which will be explored in future.

The model selection procedure achieves better results for scenarios with balanced classes. Although the models are selected on the basis of data generated by the simple Gaussian distributions, the performance of an algorithm is relatively high for all scenarios. Scores achieved for the reference concept model prove that the model representation method has an influence on the efficiency of the model selection algorithm and if the reference concept data reflects the data window distribution well, the algorithm is able to perform almost flawlessly. Other data distribution representation methods are to be evaluated in future.

Future works include:

- Introducing adaptive classification model techniques using unsupervised learning on the basis of the data properties and an information provided by the model selection algorithm,
- Designing and evaluating other data distribution representation and dataset simulation methods,
- Estimating the influence of the data dimensionality on the model selection algorithm's performance,
- Experimenting with the imbalanced class datasets and designing a method to minimize the negative influence of the imbalanced classes,
- Evaluating other concept drift detection algorithms for the model selection procedure.

Acknowledgement

This work is supported by the Polish National Science Center under a grant N N519 650440 for the period 2011-2014.

References

[A. Asuncion, 2007] A. Asuncion, D. N. (2007). UCI Machine Learning Repository.

[Alpaydin, 2010] Alpaydin, E. (2010). Introduction to Machine Learning. 2nd edition, The MIT Press.

[Attar et al., 2010] Attar, V., Sinha, P. and Wankhade, K. (2010). A fast and light classifier for data streams. Evolving Systems *1*, 199–207.

[Cliff et al., 1992] Cliff, D., Harvey, I. and Husbands, P. (1992). Incremental Evolution of Neural Network Architectures for Adaptive Behaviour.

[Dries and Rückert, 2009] Dries, A. and Rückert, U. (2009). Adaptive concept drift detection. Stat. Anal. Data Min. *2*, 311–327.

[Duda et al., 2001] Duda, R. O., Hart, P. E. and Stork, D. G. (2001). Pattern Classification (2nd Edition). 2 edition, Wiley-Interscience.

[Friedman and Rafsky, 1979] Friedman, J. and Rafsky, L. (1979). Multivariate generalizations of the Wald-Wolfowitz and smirnov two-sample tests. Annals of Statistics *7*, 697–717.

[Gama and Medas, 2005] Gama, J. a. and Medas, P. (2005). Learning decision trees from dynamic data streams. j-jucs *11*, 1353–1366.

[Greiner et al., 2002] Greiner, R., Grove, A. J. and Roth, D. (2002). Learning cost-sensitive active classifiers. Artif. Intell. *139*, 137–174.

[Hotelling, 1931] Hotelling, H. (1931). The Generalization of Student's Ratio. Annals of Mathematical Statistics *2*, 360–378.

[Hulten et al., 2001] Hulten, G., Spencer, L. and Domingos, P. (2001). Mining time-changing data streams. In Acm Sigkdd Intl. Conf. On Knowledge Discovery And Data Mining pp. 97–106,.

[Kifer et al., 2004] Kifer, D., Ben-David, S. and Gehrke, J. (2004). Detecting change in data streams. In Proceedings of the Thirtieth international conference on Very large data bases - Volume 30 VLDB '04 pp. 180–191, VLDB Endowment.

[Kuncheva, 2011] Kuncheva, L. I. (2011). Change Detection in Streaming Multivariate Data Using Likelihood Detectors. IEEE Transactions on Knowledge and Data Engineering *99*.

[Kurlej and Wozniak, 2012] Kurlej, B. and Wozniak, M. (2012). Active learning approach to concept drift problem. Logic Journal of the IGPL *20*, 550–559.

[Lane and Brodley, 1999] Lane, T. and Brodley, C. E. (1999). Temporal sequence learning and data reduction for anomaly detection. ACM Trans. Inf. Syst. Secur. *2*, 295–331.

[Lindstrom et al., 2008] Lindstrom, P., Delany, S. J. and MacNamee, B. (2008). AUTOPILOT: Simulating Changing Concepts in Real Data. In Procs. of 19th Irish Conference on Artificial Intelligence and Cognitive Science, (AICS-08), (Bridge, D., Brown, K., O'Sullivan, B. and Sorensen, H., eds), pp. 272–281,.

[Lindstrom et al., 2011] Lindstrom, P., Mac Namee, B. and Delany, S. J. (2011). Drift Detection Using Uncertainty Distribution Divergence. In Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops ICDMW '11 pp. 604–608, IEEE Computer Society, Washington, DC, USA.

[Markou and Singh, 2003] Markou, M. and Singh, S. (2003). Novelty detection: a review–part 1: statistical approaches. Signal Processing *83*, 2481–2497.

[Muhlbaier et al., 2009] Muhlbaier, M. D., Topalis, A. and Polikar, R. (2009). Learn^{++}.NC: Combining Ensemble of Classifiers With Dynamically Weighted Consult-and-Vote for Efficient Incremental Learning of New Classes. IEEE Transactions on Neural Networks *20*, 152–168.

[Narasimhamurthy and Kuncheva, 2007] Narasimhamurthy, A. and Kuncheva, L. I. (2007). A framework for generating data to simulate changing environments. In Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications AIAP'07 pp. 384–389, ACTA Press, Anaheim, CA, USA.

[Ramamurthy and Bhatnagar, 2007] Ramamurthy, S. and Bhatnagar, R. (2007). Tracking Recurrent Concept Drift in Streaming Data Using Ensemble Classifiers. In Proceedings of the Sixth International Conference on Machine Learning and Applications ICMLA '07 pp. 404–409, IEEE Computer Society, Washington, DC, USA.

[Revuz and Yor, 2004] Revuz, D. and Yor, M. (2004). Continuous Martingales and Brownian Motion (Grundlehren der mathematischen Wissenschaften). 3rd edition, Springer.

[Rubin, 1973] Rubin, D. B. (1973). Matching to Remove Bias in Observational Studies. Biometrics *29*.

[Sheskin, 2011] Sheskin, D. J. (2011). Handbook of parametric and nonparametric statistical procedure. 5th ed. 5th ed. edition, Boca Raton, FL: CRC Press.

[Smetek and Trawinski, 2011] Smetek, M. and Trawinski, B. (2011). Selection of Heterogeneous Fuzzy Model Ensembles Using Self-adaptive Genetic Algorithms. New Generation Comput. *29*, 309–327.

[Smirnov, 1948] Smirnov, N. V. (1948). Table for estimating the goodness of fit of empirical distributions. Ann. Math. Stat. *19*, 279–281.

[Sobolewski and Woźniak, 2011] Sobolewski, P. and Woźniak, M. (2011). Artificial recurrence for classification of streaming data with concept shift. In Proceedings of the Second international conference on Adaptive and intelligent systems ICAIS'11 pp. 76–87, Springer-Verlag, Berlin, Heidelberg.

[Sobolewski and Woźniak, 2012] Sobolewski, P. and Woźniak, M. (2012). Data with Shifting Concept Classification Using Simulated Recurrence. In Intelligent Information and Database Systems pp. 403–412, Springer Verlag.

[Spinosa et al., 2008] Spinosa, E. J., de Leon F. de Carvalho, A. P. and Gama, J. a. (2008). Cluster-based novel concept detection in data streams applied to intrusion detection in computer networks. In Proceedings of the 2008 ACM symposium on Applied computing SAC '08 pp. 976–980, ACM, New York, NY, USA.

[Vreeken et al., 2007] Vreeken, J., van Leeuwen, M. and Siebes, A. (2007). Characterising the difference. In KDD, (Berkhin, P., Caruana, R. and Wu, X., eds), pp. 765–774, ACM.

[Wang et al., 2005] Wang, P., Wang, H., Wu, X., Wang, W. and Shi, B. (2005). On Reducing Classifier Granularity in Mining Concept-Drifting Data Streams. In Proceedings of the Fifth IEEE International Conference on Data Mining ICDM '05 pp. 474–481, IEEE Computer Society, Washington, DC, USA.

[Wang et al., 2011] Wang, S., Schlobach, S. and Klein, M. C. A. (2011). Concept drift and how to identify it. J. Web Sem. *9*, 247–265.

[Wilcoxon, 1945] Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. Biometrics Bulletin *1*, 80–83.

[Yamauchi, 2010] Yamauchi, K. (2010). Incremental model selection and ensemble prediction under virtual concept drifting environments. In Proceedings of the 11th Pacific Rim international conference on Trends in artificial intelligence PRICAI'10 pp. 570–582, Springer-Verlag, Berlin, Heidelberg.

[Zhou et al., 2009] Zhou, J., Fu, Y., Wu, Y., Xia, H. and Fang, Y. (2009). Anomaly Detection over Concept Drifting Data Streams. Journal Of Computational Information Systems *6*, 16971703.

[Zliobaite, 2008] Zliobaite, I. (2008). Expected Classification Error of the Euclidean Linear Classifier under Sudden Concept Drift. In Proceedings of the 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery - Volume 02 FSKD '08 pp. 29–33, IEEE Computer Society, Washington, DC, USA.

[Zliobaite, 2010] Zliobaite, I. (2010). Change with Delayed Labeling: When is it Detectable? In Proceedings of the 2010 IEEE International Conference on Data Mining Workshops ICDMW '10 pp. 843–850, IEEE Computer Society, Washington, DC, USA.

[Zliobaite and Kuncheva, 2009] Zliobaite, I. and Kuncheva, L. I. (2009). Determining the Training Window for Small Sample Size Classification with Concept Drift. In Proceedings of the 2009 IEEE International Conference on Data Mining Workshops ICDMW '09 pp. 447–452, IEEE Computer Society, Washington, DC, USA.