Survey on Concept Drift

Jan Knotek and Welma Pereira

Faculty of Economics - University of Porto, Portugal

Abstract. A wide range of applications often have to deal with datasets from non-stationary distributions. In this cases the learning task is more challenging. The learning problem of data from distributions that change over time is known as concept drift. The aim of this document is to describe this problem and to provide a summary of the main existing approaches for dealing with concept drift in data streams.

1 Introduction

The machine learning algorithms assume that the instances are independent and identically distributed generated by some probability distribution. In real-world applications this methods are often applied to datasets from dinamic environments that may change over time. One example where this happens is the weather prediction that can change more or less radically with phenomenons that depend on the climatic variations of different places. Another example is the patterns of customer's buying preferences that may change with time, depending on the current day of the week, availability of alternatives, inflation rate, etc. Let's say we want to predict weekly merchandise sales, and we have developed a predictive model that works to our satisfaction. The model may use inputs such as the amount of money spent on advertising, promotions you are running, and other metrics that may affect sales. What we are likely to experience is that the model will become less and less accurate over time. This can happens because the distribution underlying the data is likely to change over time. The problem of learning from data from distributions that change over time is known as concept drift. The term concept refers to the quantity you are looking to predict. More generally, it can also refer to other phenomena of interest besides the target concept, such as an input, but, in the context of concept drift, the term commonly refers to the target variable.
When concept drift occurs, it means that the instances are no longer from the same distribution. This has important consequences because most of the learning theoretic performance guarantees used in machine learning are based on this assumption. The meaning is that when concept drift occurs the performance of most learning algorithms becomes less accurate as the times passes. The learning model underlying successful predictions should be able to adapt accordingly with some regular updating.
This papers is organized as follows: section 2 we give important definitions, describe the types and causes of changes. Section 3 contains main approaches for dealing the problem. Section 4 is explaining the principes of some selected methods for dealing with concept drift online in data streams.

2 Definition of Source

We can define a classification problem, independently of the presence of concept drift, as [1]:
Let $X \in \Re^p$ is an instance in p-dimensional feature space. $X \in c_i$, where $c_1, c_2, ..., c_k$ is the set of class labels. The optimal classifier to classify $X \rightarrow c_i$ is completely determined by a prior probabilities for the classes $P(c_i)$ and the class-conditional probability density functions (pdf) $p(X/c_i)$, i = 1,...,k. We define a set of a prior probabilities of the classes and class-conditional pdf's as concept or data source:

$$S = (P(c_1), p(X_j c_1)), (P(c_2), p(X_j c_2)), ..., (P(c_k), p(X_j c_k)) \quad (1)$$

When referring to a particular source at time t we will use the term source, while when referring to a fixed set of prior probability and the classes and class-conditional pdf we will use the term concept and denote it S.

2.1 Types of Change

Usually concept drift can occur in the following ways. For simplification let's consider that there are just two sources for the examples S_1 and S_2:

- **Sudden** (also called Concept Shift) : when at time t0 a source S_1 is suddenly replaced by some source S_2 . For instance, a peak in sales of ice cream is associated with summer but it can start at different time every year depending on the temperature and other factors like climate change. It is not known when the peak can start and therefore provoke a sudden change in the ice cream sales;
- **Gradual**: there are two types under this term. The first type of gradual drift is referring to a period when both sources S_1 and S_2 are active. As time passes, the probability of sampling from S_1 decreases, probability of sampling from S_2 increases so as at some point just instances from source S_2 can be seen;
- **Incremental**: also referred as gradual includes more than two sources, however, the difference between the sources is very small, thus the drift is noticed only when looking at a longer time period. For example the interested in political news tends to increase as the elections are approaching;
- **Reoccurring**: when previously active concept reappears after some time. It is not the same as seasonality effect because is not periodic for sure, it is not known when the source might reappear. Some rare weather phenomenons like hurricanes might reoccur in certain places.

Note that the above discussed types of drifts are not exhaustive. If we think of a data segment of length t and just two data generating sources S_I and S_2 , the number of possible combinations of the distributions would be 2^t, which means quite a lot possible change patterns. Moreover, in concept drift research it is often assumed that the data stream is endless, thus there could be infinite number of possible change patterns. The descriptions given here it is just a help on designing strategies to deal with concept drift.

2.2 Causes of Change

The causes of the changes can be due to modifications in the context of learning due to changes in hidden variables or it can happens because of changes in the characteristic properties of the observed variables.
Whenever a change in the underlying concept generating data occurs, the class-distribution of examples changes.
A virtual drift is characterized when there are changes in the class-distribution without concept drift.
The change may occur in thee ways:

- Class priors $P(c)$ might change over time;
- The distributions of one or several classes $p(X/c)$ might change;
- The posterior distributions of the class memberships $p(c/X)$ might change.

The concept drift is usually characterized when there are gradual changes in the conditional distribution of the label $p(c/X)$.

3 Some Approaches to handle concept drift

The authors of the term "concept drift" are Schlimmer and Granger that in 1986 formulated the problem of incremental learning from noisy data and presented an adaptive learning algorithm STAGGER [15]. The STAGGER was the first concept drift handling system. It maintains a set of concept descriptions, which are originally features themselves, and more complicated concept descriptions are then produced iteratively using feature construction, the best of which are selected according to their relevance to the current data. After that many studies dealing with concept drift problem appeared. In the following we intended to give a general view of some of the main approaches to handle concept drift.

3.1 Instance Selection

The goal of the approach instance selection is to select instances relevant to the current concept. The most common concept drift handling technique is based on instance selection and consists in generalizing from a window that moves over recently arrived instances, as new examples arrive they are inserted into the beginning of the window, a corresponding number of examples is removed from the end of the window, and the learner is reapplied. The learnt concepts are used for prediction only in the immediate future.
Some algorithms use a window of fixed size [11], while others use heuristics to adjust the window size to the current extent of concept drift. The window of fixed size is a fast and easy to implement solution, but it requires a preliminary investigation of the domain to select the window size.
One of the most known system that use this approach is the FLORA family of algorithms.

FLORA Family of Algorithms: FLORA is a supervised incremental learning system that takes as input a stream of positive and negative example of a target concept that changes over time. The first FLORA algorithm uses a fixed moving window approach to process the data. The concepts are stored and the update process involves two processes: a learning process (adjust concept description based on the new data) and a forgetting process (discard data may be out of date). FLORA2 was introduced to address some of the problems associated with FLORA such as the fixed window size. FLORA2 has a heuristic routine to dynamically adjust its window size and uses a better generalization technique to integrate the knowledge extracted from the examples observed. The algorithm was further improved to allow previously extracted knowledge to help deal with recurring concepts (FLORA3) and to allow it to handle noisy data (FLORA4).
Lazaresc et al. presented in 2003 a multiple window incremental unsupervised learning algorithm to deal with concept drift [10]. It uses three windows of different sizes to estimate the change in the data. This approach aims at dealing more effectively with different types of drift (the system's adaptation is more progressive).

3.2 Instance Weighting

The idea behind instance weighting is of a gradual forgetting. This means that newer training examples should be more important than older ones and their importance should decrease with time. The importance of example is given with its weights. The weights can be calculated using some gradual forgetting function. A kernel function can also be used for this task. SVMs and neural networks have been used for that. The Instances can be weighted according to their age, and their competence with regard to the current concept.

3.3 Ensemble Learning

Ensemble learning or learning with multiple concept descriptions, maintains a set of concept descriptions, predictions of which are combined using voting or weighted voting, or the most relevant description is selected. The weight is usually a function of the historical performance and indicates the'competence' of a base learner, expected in the future. For example, one way of using this approach is that: if there is a change in the environment, the individual classifiers are re-evaluated and the worst classifier is replaced by a new one trained on the most recent data. Wang et al. [17] propose to evaluate all classifiers using the most recent "chunk" of data as the testing set. The classifiers whose error exceeds a certain threshold are discarded from the ensemble.
For dealing with large datasets the simple models are preferred because there might not be time for running and updating an ensemble. However, when time is not so important and high accuracy is required, an ensemble can be a very good solution.

3.4 Statistical Methods

Given a sequence of training examples, are the last n_1 examples sampled from a different distribution than the n_2 preceding ones?
These methods typically compute a statistic that catches the similarity between two example sets of a multivariate data. The value of the statistic is then compared to the expected value under the null hypothesis that both sets are sampled from the same distribution. The resulting p-value can be seen as a measure of to what extent concept drift has happened. One of first studies by Friedman and Rafsky [4] extended the Wald-Wolfowitz and the Smirnov tests towards the multivariate setting. Some later studies use nearest neighbor methods to compute the statistic [12], and some others require a complete matrix of dissimilarity measures between all examples as determined by a kernel [2].

The categories presented here are not exhaustive and they have been used combined with each other in different ways. More information can be found in [5] [18] [16].

4 Examples of online learning methods

4.1 Drift Detection Method

Drift detection method was introduced by João Gama[3]. It is an online learning method, which uses classifier to predict coming data.
Data are represented by pairs $< \vec{x}, y >$, where $\vec{x}$ is a vector of attributes and y is a class. Model is making a prediction y' for class y. If $y' = y$, prediction was correct, otherwise error appeared. Both examples are used to learn classifier.
Supposing that data have same distribution all the time, errors in classification are appearing with binomial distribution with probability p_i. This probability can be called error rate and its standard deviation s_i is defined:

$$s_i = \sqrt{p_i(1 - p_i)/i} \tag{2}$$

As the classifier become learned for the coming data, error rate becomes to be stable. This value is saved as p_{min} together with its standard deviation s_{min}.
Change in distribution of data, know as concept drift, is shown by increase in error rate p_i. For detecting such change, two levels for the error rate are defined:

1. Warning level – beyond this level, the examples are stored in anticipation of a possible change of context.

$$s_i + p_i \geq p_{min} + \alpha \cdot s_{min} \tag{3}$$

2. Drift level – the learned model is reset and a new model is learnt using the examples stored since the warning level triggered.

$$s_i + p_i \geq p_{min} + \beta \cdot s_{min} \tag{4}$$

Values for α, β are set for desired confidence level, admissable is $\alpha = 2, \beta = 3$.
Method performance is good for abrupt changes, where there is visible quick change in classifier error rate, but slowly gradual changes are detected too late.

4.2 Early Drift Detection Method

Slightly different method was presented by Baena-García[6]. Purpose of creating Early Drift Detection Method was to improve sensitivity to slowly gradual drifts which was detected too slowly by previous method. Classifier is used for predicting of coming data too. EDDM is using average error distance p'_i instead of error rate to detect change. Standard deviation s'_i is defined:

$$s'_i = \sqrt{p'_i(1 - p'_i)/i} \tag{5}$$

When $p'_i + 2 \cdot s'_i$ reaches its maximum (average error distance is maximal) variables p_{max} and s_{max} are stored. Point $p'_{max} + 2 \cdot s'_{max}$ shows moment, when the classifier is representing the data in a best possible way. Then two levels for warning and drift are defined:

1. Warning level – beyond this level, the examples are stored in advance of a possible change of context.

$$(p'_i + 2 \cdot s'_i)/(p'_{max} + 2 \cdot s'_{max}) < \alpha \tag{6}$$

2. Drift level – concept drift is supposed to be true, the model induced by the learning method is reset and a new model is learnt using the examples stored since the warning level triggered.

$$(p'_i + 2 \cdot s'_i)/(p'_{max} + 2 \cdot s'_{max}) < \beta \tag{7}$$

Due to use of fraction in the definition of levels, values for α, β are different. It was set after experimentation to $\alpha = 0.95, \beta = 0.90$.
EDDM is much more sensitive than DDM, but this can be taken as a disadvantage, when data are noisy.

4.3 Online Tree, Online Tree2

OnlineTree (Nuñez et al., 2005)[13] is able to detect concept drift from small data sets (less than 200 examples) and manage noise level in data, but it does not work when the data set has numerical features, a data stream is present, change in noise levels appears and/or the problem contains virtual drift.
OnlineTree2 [14] corrects these deficiencies, being able to deal with data streams containing unknown dynamics (possible concept drifts, changes in noise level, virtual drift, continuous or symbolic features and different distribution of examples).
Better behavior is achieved by using local adaptive windows with a new strategy which forgets examples as a result of the leaf reducing its window size when the local performance decreases.

Experimentation shows that OnlineTree2 achieves low error rates, improves the number of stored examples and has a reduced processing time. Problem with this algorithm is its efficiency. Where CVFDT takes about 2000 instances per second, OnlineTree2 can take only 30 instances per second.

4.4 Multi-Resolution Learning

The algorithm from 2005 created by Mihai M. Lazarescu[9] attempts to interpret current data as well as detect, predict and quickly adapt to future changes in the concept.
The work presented makes three novel contributions. The first novel aspect of the research is that the algorithm uses a usefulness based approach to control the forgetting mechanism used to discard data from the system's memory.
The forgetting mechanism analyses the newly observed data instance to determine how well it fits with the rest of the instances in the system's memory and the current concept definition. The mechanism also determines if the new instance is a likely indicator of a change in the concept. Usefulness and age of the instance is used to determine whether or not the instance is to be discarded.
The second contribution is that the algorithm predicts the rate of change to give the system a pro-active approach to the data and thus improve the accuracy of the concept tracking. Information about the rate of change is used to improve the control over the size of the larger dynamic data window.
The third contribution is the representation used for the concepts tracked. The concept is represented through a combination of instance generalizations, data predictors and the rate of change observed when concept was stable. Old concept descriptions are stored in knowledge repository to avoid having to "re-learn" recurrent concepts.
Algorithm consists of 3 modules: Evidence Based Forgetting module, Prediction Analysis Module and algorithm itself.
Overall algorithm uses two windows which compete to produce the best interpretation of data. Reason for this is to deal with all types of drift. Windows have different sizes which allows higher flexibility – instance forgot in smaller window is kept in larger window.
At each step t, instance generalizations, data predictors and the rate of change are estimated for both window. The concept definition is updated by formula $\bar{x}_t(w) = 1/|w| \sum_{i=0}^{|w|-1} x_{t-i}$.
This newly defined concept is used to compute the change in window at time $t-1$ from the concept value at time t. Current change is compared to consistency threshold ϵ_c. If it is lower, then the persistence value for the window p, is incremented by one. Else the persistence value of the window is reset to 1.
Consistency and persistence are the parameters of concept drift. If the change observed in the target concept is both consistent and persistent then the drift is considered to be permanent. If the change is consistent but not persistent then the drift is considered to be virtual. Finally, if

the change observed is neither consistent or persistent then the drift is considered to be noise.
Then algorithm analyses the data samples stored in the window to determine their usefulness as well as possible forgetting points (ϕ). Values with low evidence are added to forget set (ψ). After computing the rate of change, instances in ψ set are deleted and window size is adjusted.
Two parameters are set by user. The length of history to be used when analyzing the trend of change in the concept–default is $\frac{w}{2}$ which provides good results in all experiments in paper. Second parameter is the delay used to prevent the removal of instances with low usefulness, but are likely indicators of change.

4.5 Tracking Drifting Concepts by Time Window Optimization

In paper presented by Ivan Koychev and Robert Lothian in 2005 [8] window optimization method is proposed. Authors are using the Golden section algorithm to trap the best window size with highest accuracy.
The paper presents a mechanism for dealing with the concept drift problem which uses a statistical test to detect whether the current concept is changing. If a concept drift is detected, then the mechanism optimizes the time window size to achieve maximum accuracy of prediction.
Main contribution is, that the proposed optimization is independent on a learning algorithm and it is significantly increasing accuracy. The algorithm is self-adapting and it can be used in many datasets without any predefined domain-dependent heuristics or parameter.

4.6 STEPD: Detection Method Using Statistical Testing

Method presented by K. Nishida, K. Yamauchi [7] in 2007 uses a statistical test of equal proportions to detect various types of concept drift.
Authors aim was to deal with problems of Drift detection method and Early drift detection method. Firstly multiple classifier system was developed. After simplification new method with one online classifier called ACED was introduced.
ACED observes the predictive accuracy of the online classifier for recent W examples, q_t, and calculates the $1 - \alpha_d$ confidence interval for q_t at every time t.
ACED presumes, that q_t will not fall below the lower endpoint of the interval at time $t - W$, q^l_{t-W}, if the target concept is stationary. Thus,it initializes the classifier if $q_t < q^l_{t-W}$. Note that it starts detecting drift after receiving $2W$ examples.
Method STEPD basic principle is to use two accuracies - recent accuracy and overall one. Authors assume two things: the accuracy of a classifier for recent W examples will be equal to the overall accuracy if the target concept is stationary; and a significant decrease of recent accuracy suggests that concept is changing.
The test is performed by calculating following statistics:

$$T(r_o, r_r, n_o, n_r) = \frac{|r_o/n_o - r_r/n_r| - 0.5(1/n_o + 1/n_r)}{\sqrt{\hat{p}(1-\hat{p})(1/n_o + 1/n_r)}} \tag{8}$$

Where r_o is the number of correct classification among the overall n_o examples except W examples, r_r is the number of correct classifications among the W $(= n_r)$ examples and $\hat{p} = (r_o + r_r)/(n_o + n_r)$.
Value is compared to the percentile of the standard normal distribution to obtain the observed significance level (P-value).
If the P-value, P, is less than a significance level, then $r_o/n_o > r_r/n_r$ (overall accuracy is higher than recent accuracy) and the concept drift is detected.
STEPD uses two significance levels: α_w and α_d. Examples are stored in a short-term memory while $P < \alpha_w$. Classifier is rebuilded from stored examples and all variables are reset if $P < \alpha_d$.
Detecting starts when $n_o + n_r \geq 2W$ and the stored examples are removed if $P \geq \alpha_w$.
Results presented on 5 synthetic datasets shows, that STEPD has the best results for abrupt changes from compared methods (DDM, EDDM, ACED). Results for gradual changes are comparable to EDDM. ACED has too much misclassification and DDM is two slow in detection.

5 Conclusion

There is no universal approach to deal with concept drift. The techniques to be used depend in the properties of the problem. The approaches are usually create based in problems with artificially created datasets. It is difficult to rate the methods by artificial dataset because they can perform differently in real problems. Therefore there is a necessity for a concept drift and algorithm database. Using this database user can match the algorithm, that suits the real problem, based on statistics of the performances of algorithms and on the properties of artificial datasets.
Recently many algorithms exist. Some of these algorithms are better to deal abrupt drifts, some are more sensitive to gradual changes. To deal with real problems, which contain all kinds of drifts combination of these techniques can be used with better results.

References

1. L.I.Kuncheva A.Narasimhamurthy. A framework for generating data to simulate changing environments, 2007.
2. Malte Rasch Bernhard Scholkopf Arthur Gretton, Karsten M. Borgwardt and Alexander J.Smola. A kernel method for the two-sampleproblem. 2006.
3. Gama et. al. Learning with drift detection, 2004. Lecture Notes in Computer Science 3171.
4. Jerome H. Friedman and Lawrence C. Rafsky. Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests. *Annals of Statistics*, pages 697–717, 1979.
5. Jing Jiang. A literature survey on domain adaptation of statistical classifiers, 2007.

6. Manuel Baena-Garca Jose, José Del Campo-Ávila, Raúl Fidalgo, Albert Bifet, Ricard Gavalda, and Rafael Morales-Bueno. Early drift detection method.
7. K. Yamauchi K. Nishida. Detecting concept drift using statistical testing, 2007. in Proceedings of the Tenth International Conference on Discovery Science (DS07) - Lecture Notes in Artificial Intelligence.
8. I. Koychev and R. Lothian. Tracking drifting concepts by time window optimization, 2005. Development in Intelligent Systems XXII Proceedings of AI-2005, the Twenty-fifth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence.
9. Mihai M. Lazarescu. A multi-resolution learning approach to tracking concept drift and recurrent concepts.
10. Mihai M. Lazarescu, Svetha Venkatesh, and Hung H. Bui. Using multiple windows to track concept drift. In *In Intelligent Data Analysis Journal*, 2003.
11. Tom Mitchell, Rich Caruana, Dayne Freitag, John McDermott, and David Zabowski. Experience with a learning personal assistant, 1994.
12. Tajvidi N. Permutation tests for equality of distributions in high-dimensional settings. *Biometrika*, pages 89:359–374, 2002.
13. Marlon Núñez, Raúl Fidalgo, and Rafael Morales. On-line learning of decision trees in problems with unknown dynamics, 2005. In Proceedings of the 4th Mexican International Conference on Artificial Intelligence.
14. Marlon Núñez, Raúl Fidalgo, Rafael Morales, and Claude Sammut. Learning in environments with unknown dynamics: Towards more robust concept learners, 2007.
15. Jeffrey C. Schlimmer and Jr. Richard H. Granger. Incremental learning from noisy data. *Machine Learning*, 1:317–354, 1986.
16. Alexey Tsymbal. The problem of concept drift: Definitions and related work. Technical report, 2004.
17. Haixun Wang, Wei Fan, Philip S. Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers, 2003.
18. Gerhard Widmer. Learning in dynamically changing domains: Recent contributions of machine learning.