* Pages – 8/9 seems to be the sweet spot
* Total word count – 5000 – 6500
* Introduction – including abstract, 1 – 1.5 pages (600 – 1000 words)
  + C
* Related work – 0.5 – 1 pages (400 – 800 words)
  + X
* Methodology (1.5 – 3 pages)
  + Problem definition
    - X
  + Adaptation technique
    - X
* References – 20 - 25

# Detecting and Adapting to Concept Drift in Continually Evolving Stochastic Processes

Upeka Premaratne

University of Moratuwa, Sri Lanka

upeka@ent.mrt.ac.lk

Sunanda Gamage

rcsunanda@gmail.com

**Abstract**

[to fill]

**Keywords**: concept drift, drift detection, drift adaptation, incremental learning, ensemble methods, machine learning

## 1. Introduction

The vast majority of machine learning research focuses on building models (eg: regression models, classifiers) that are static; i.e. the model is trained one time on a training dataset and then applied to test data. This works fine when the underlying stochastic process that is modeled remains stationary. However, in many real world applications, the process is non-stationary, and it continuously changes with time.

For example, consider the video feed from a traffic camera, which can undergo changes from morning to night, and from summer to winter. The video feed process in this example undergoes gradual continuous change. The drift can also be abrupt and large, such as an impactful market event that affects stock prices.

The underlying stochastic process in these examples is non-stationary, which means that the joint probability distribution of the process attributes,, is time varying. This is known as concept drift. Applying a model that is trained one time with some initial training dataset to such a process results in performance degradation. Therefore, methods to characterize the concept drift, and adapt machine learning models to drift by incremental learning are an important research direction in machine learning.

[Summarize current methods and their deficiencies ]

In this work, we improve on the drift detection techniques used in [A] and [B], and present an efficient ensemble technique of incremental learning to model a continuously changing stochastic process. The improved drift detection technique tracks a difference metric (what metric?) between probability distributions estimated from two sample windows before and after a time point. The estimated distribution is typically the joint probability distribution of attributes for the unsurprised case, or joint attribute and class distribution for the supervised case. When the continuous sum of this metric since the last adaptation rises above a threshold , it is decided that the process has drifted and an adaptation should be made to the learned model.

At this point, a new model is learned from a suitable number of most recent samples, and it is added to an ensemble of previously learned models. The final time varying model is a weighted some of the sub models in the ensemble. The weights can be set in a way such that older sub models have less relevance (a form of forgetting). They can also be set to reflect the similarity between the probability distributions of the current sample window, and the windows corresponding to each sub model. In the latter case, the distributions relevant to each sub model also need to be stored.

Our drift detection and adaption techniques can act as a wrapper method independent of the machine learning model being employed. The drift detection technique can be made to detect different types of drift [B] by choosing the relevant distribution for computing the difference metric. The methods work well for both supervised learning models such as classifiers, or unsupervised models such as hidden Markov models. The adaptation technique does not cause catastrophic forgetting. The forgetting factor can in fact be controlled by setting appropriate weights on the sub models in the ensemble. Experiment results of evaluating the methods on artificial and real-life datasets show good performance in detecting drift and adapting to it.

xx

The rest of this paper is organized as follows. Section 2 presents related work in detecting concept drift and adaptation methods. In Section 3, the drift problem is formally defined, and solutions for drift detection and adaptation are described. In Section 4, we present the results of evaluating the method on several artificial datasets and a real-life dataset. Some important implementation details and possible optimizations are described in Section 5. Conclusions and future directions are given in Section 6.

## 2. Related Work

### 2.1 Detection techniques

Several works in the literature have developed drift detection techniques. In [A], drift is measured by tracing the error rate at test time, where a rise in error is considered as a sign of drift. Two threshold levels, warning level and drift level, are predefined, and when error increases above the drift level, the detector triggers an alarm. This is method is computationally efficient, but needs the true labels at test time to compute the error rate.

A novel method introduced in [B], called concept drift maps, computes a difference metric in real-time between probability distributions (Total Variation Distance) estimated from two sample windows before and after a time point. This method allows the use of any probability distribution that represents the dataset, including the distribution of any subset of attributes. Using a distance metric instead of test error rate means it is well suited for cases where true labels are not revealed at test time.

### 2.2 Adaptation techniques

There is a large body of work that explores adaptation methods for concept drift. An excellent overview of these methods and their key design principles is given in [C] and [N]. A common approach is to divide the historical data into blocks or windows that represent a context (a period where no drift has occurred or the process is stable) and use it to train a learner [D][E]. Alternatively, training sets can be formulated by instance weighing, where weights are assigned according to the age of instances or their performance with regard to the current concept [G][H].

The adaptation method described in [A] retrains a new model on the most recent context when a drift is detected, and any previous model is discarded. This leads to catastrophic forgetting, which means that any previously learned knowledge is completely forgotten, and recurring patterns will have no benefit from it.

In order to retain previous knowledge, [E][F] use an ensemble of models. In [F], Sequentially arriving samples are batched together by a fixed batch size, and weights of previous sub models in the ensemble are updated based on their performance on the batch. Then, the batch is sampled to extract the instances that were not classified accurately by previous sub models, and a new sub model is trained on them. This method does not detect drifts, but keeps updating the ensemble weights and adding new sub models as new batches arrive. This is computationally not efficient. Furthermore, this technique requires that true labels of new examples must be available.

Fields that are related to learning under concept drift are “domain adaptation” and “transfer learning”. In these areas, the topic of adapting a model trained on one domain or task to a different one is studied. The majority of this work considers one source domain and one or a few target domains. However, some researchers have extended domain adaptation to the incremental setting [J][K][L]. In [J], a method to adapt a base classifier to a change in class probability is described. The authors also present an extension for the case of sequentially arriving examples. However, this method does not work for changes in distributions of attributes or class conditional attributes.

**[Our unique points, in comparison to above – elaborate on these]**

* Especially good for generative models (unsupervised), since we have change detection of P(X)
  + Weight updates is made possible by using distance metric rather than classification error
* Unlike [F], we have a drift detector and it can detect multiple types of drift
* Because ours is an ensemble method, and due to distance based weighting of sub models,
  + Previously learned knowledge is not discarded
  + Different sub models can master different parts of the data distribution
  + works well for recurring concepts
* First known use of concept drift maps
* Yields well to a practical implementation and optimizations (

## 3. Methodology

### 3.1 Problem definition

Consider a stochastic process characterized by a joint distribution over the random variables . If our task is classification or regression, a target variable is also present, and the process is represented by the joint distribution over . A sequence of samples generated by this process are observed.

We say a concept drift has occurred in the process between two time points when . Since it is often not practical to estimate the distribution in effect at a specific point in time, we consider drift occurring between time intervals and , which is characterized by .

The drift detection problem is to quantify the above change in probability distributions with time, and identify when significant drift has occurred. The adaptation problem is to make necessary changes to a learned model so that it can be applied on the process and results in minimal error.

### 3.2 Detection technique

For drift detection, we combine and improve on the key ideas presented in [A] and [B]. To quantify drift, we continuously compute a difference metric between the probability distributions and as illustrated in Figure 1. can be any set of variables that captures the drift in the process, such as or or . In this work, we use the metric Total Variation Distance [M] (or any other metric? Why).

Figure 1: Time windows for drift measurement

samples

is tracked over time, and values above some low threshold are summed. The purpose of having the low threshold is to prevent noise or other artifacts from affecting the cumulative sum and triggering false alarms. When the difference metric is rising, and the cumulative sum increases above a certain threshold, a drift is detected and reported.

The variable used for the distance metric calculation will affect what type of concept drift will be captured by the detector. captures drift in its full form while the drift reflected by is known as virtual drift. captures drift due to class imbalance. The above choices of require that true class labels of test data instances be revealed to the detector (supervised drift detection). In the unsupervised case, can be used to efficiently capture concept drift, and we use this choice of in our detection algorithm.

### 3.3 Adaptation technique

We propose an adaptation technique that employs an ensemble of weighted sub models (base learners) in this work. Initially, the ensemble has only the originally learned sub model. When a drift is detected by the detection method, the following actions are taken

1. A new sub model is trained from the examples in the current window and added to the ensemble.
2. Weights of sub models in the ensemble are updated based on the distance between the current distribution and the distribution of the window on which the sub model was trained.

The weight of a sub model represents the similarity between the current distribution of the process, and the distribution that existed at the time of training the sub model. Since the final model is made of the weighted sum of the sub model outputs, this ensures that sub models that are most relevant to the current context of the process are given higher weights.

The sub models in the ensemble can be any type of learner, and the adaptation algorithm imposes no constraint on what it can be. However, if it is a supervised learner, true class labels of test data instances be must be revealed in ordered to train a new sub model (supervised adaptation). On the other hand, unsupervised base learners like Hidden Markov Models and Gaussian Mixture Models can be trained without the true labels test data.

## 4. Experimental Evaluation

The proposed drift detection and adaptation technique was evaluated against artificial datasets and a real-world dataset that is commonly used in concept drift research. Both datasets represent classifier tasks, and we use a neural network with a non-linear combination of features as the classifier. The adaptation problem is a supervised classifier adaptation; i.e. the test data labels are revealed after a prediction is made on them.

The proposed drift detection and adaptation method was compared with the following baselines.

* No adaptation – original model only
* Model trained on all available data
* Model trained on only the last batch
* Exhaustive search – look for a subset of data that gives best accuracy on test set

The exhaustive search is not feasible on a data stream in practice, but it gives a lower bound for comparison purposes. All tests were repeated five times, and the average result is presented.

### 4.1 Results on artificial dataset

The artificial dataset consists of two features and a binary class variable. The class conditional probability distributions are Gaussians with an identity covariance matrix. To simulate three different scenarios of drift, the following three variations of the dataset were used.

Abrupt drift – Initially the means are and . Simulate abrupt drift by setting and in the middle of the data stream.

Gradual drift – Initially the means are and . Simulate gradual drift by moving the distributions within a period of time to and

Recurring context – The distributions are similar to the abrupt drift scenario, except that the two sets of distributions are switched at time intervals, causing two contexts to recur in the data stream.

Three data streams simulating the above three drift scenarios were given to our detection and adaptation method. Figure 2 shows the concept drift maps and illustrates where the actual drift occurred, and where the detection algorithm has triggered detections. Figure 3 compares the online error rates (calculated considering a small moving window) of our adaptation method with other baselines. The total average error results of these experiments are given in Table 1.

[results analysis, comparison with [A][F]]

### 4.2 Results on real life dataset

The real-world dataset we have used to verify our methods is the Australian New South Wales Electricity Market dataset (ELEC), which has also been used in other concept drift research [A][P][Q]. The dataset contains 45312 labeled examples collected from the Australian electricity market from 7 May 1996 to 5 December 1998. Each example on the dataset has 5 features, the day of week, the time stamp, the NSW electricity demand, the Vic electricity demand, the scheduled electricity transfer between states and the class label The class label identifies the change of the price related to a moving average of the last 24 hours.

The classifier employed in this experiment was also a neural network trained with stochastic gradient descent. Initially, the model was trained with 10000 examples, and the remaining examples were given to the detection and adaptation algorithm as a data stream.

Figure 4 shows the concept drift map and the locations where the detection algorithm has detected drifts. Figure 5 compares the online error rate of the algorithm and other baselines on this dataset. The total average error results are given in Table 2.

[results analysis, comparison with [A]]

## 5. Implementation and Optimizations

In the detection algorithm, the method used for estimating the probability distribution of a sample window is kernel density estimation (KDE), as it is a non-parametric method suitable for estimating complex high dimensional distributions. When calculating the Total Variation Distance between two KDE estimated distributions, equation (1) becomes;

For high dimensional data, this numeric integration can be done efficiently with Monte Carlo Integration.

It must be noted that in a system with real-time performance expectations, the detection and adaptation algorithms must run in parallel with the main classification task of the ensemble. This ensures that the computations needed for tracking the concept drift map in the detection algorithm do not affect the main classification task.

We also suggest several potential optimizations to our algorithms. Since our detection method uses the Total Variation Distance between two sample windows, there is a delay between when the actual drift occurs and when it is detected (this delay can be controlled by tuning the sample window size). The online error rate during this period is high, as seen in Figure 3 and Figure 5. In the supervised case (class labels of test instances are revealed to the detection algorithm), the revealed error rate could be incorporated into the detection algorithm, which will enable quick drift detection. However, this optimization is not possible in the unsupervised detection case.

Another potential optimization in the supervised case is to modify the sub model weights in adaptation (equation (2)). Since the true class labels are available, the weights of each sub model could be determined by taking into account how well each sub model performs on the latest data batch.

In this work, we have used to compute the concept drift maps in order to capture drifts in the data distribution. It is also possible to use the distribution of any subset for this purpose. By identifying subsets of features that have caused the drift, it is possible to train the new sub model of the ensemble with only those features. This will result in a better performing model with low variance, especially in the case of high dimensional datasets. However, this will introduce high computational complexity, and an efficient technique must be employed to select the subsets used in computing the concept drift maps.

## 6. Conclusion

[to fill]

[a note about how our system performs on following desirable properties

* (1) quickly adapt to concept drift
* (2) be robust to noise and distinguish it from concept drift
* (3) recognize and treat recurring contexts.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Gradual drift | Abrupt drift | Recurring context |
| Our method – window size = x |  |  |  |
| Our method – window size = y |  |  |  |
| Our method – window size = z |  |  |  |
| No adaptation |  |  |  |
| All examples |  |  |  |
| Last n batches |  |  |  |

Table 1: Total average error on artificial dataset with different adaptation methods and drift types

|  |  |
| --- | --- |
|  | Gradual drift |
| Our method – window size = x |  |
| Our method – window size = y |  |
| Our method – window size = z |  |
| No adaptation |  |
| All examples |  |
| Last n batches |  |

Table 2: Average error on real-world dataset with different adaptation methods (can switch rows/ columns of this table)

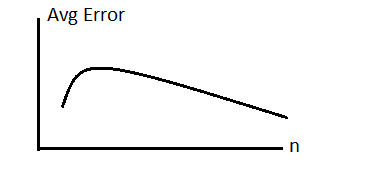
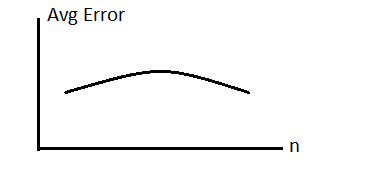


Figure 2: Left: Average error on gradual drift scenario. Right: Average error on abrupt drift scenario (plot both our method and no adaptation/ all examples)

## References

[A] Learning with Drift Detection – J. Gama

[B] Understanding Concept Drift - Geoffrey I. Webb

[C] Learning under Concept Drift: an Overview - Indre Zliobaite, 2010

[D] N. Street and Y. Kim. A streaming ensemble algorithm (sea) for large-scale classification

[E] H. Wang, W. Fan, P. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers

[F] An Ensemble Classifier for Drifting Concepts - Ralf Klinkenberg

[G] I. Koychev. Gradual forgetting for adaptation to concept drift

[H] P. Zhang, X. Zhu, and Y. Shi. Categorizing and mining concept drifting data streams

[J] Classifier Adaptation at Prediction Time - Christoph H. Lampert

[K] Continuous Manifold Based Adaptation for Evolving Visual Domains

[L] E. Levinkov and M. Fritz. Sequential Bayesian model update under structured scene prior for semantic road scenes labeling

[M] Levin D, Peres Y, Wilmer E (2008) Markov Chains and Mixing Times. American Mathematical Soc., URL <https://books.google.com.au/books?id=6Cg5Nq5sSv4C>

[N] The problem of concept drift: definitions and related work - Alexey Tsymbal

[P] Michael Harries. Splice-2 comparative evaluation: Electricity pricing. Technical report

[Q] Learning from Time-Changing Data with Adaptive Windowing (ADWIN) - Albert Bifet Ricard Gavald`a