# Analyzing the Impact of Feature Drifts in Streaming Learning

**Conference Paper** · November 2015

DOI: 10.1007/978-3-319-26532-2_3



**3 authors:**

Jean Paul Barddal
Pontifícia Universidade Católica do Paraná (PUC-PR)
**57** PUBLICATIONS **880** CITATIONS
SEE PROFILE

Heitor Murilo Gomes
The University of Waikato
**58** PUBLICATIONS **968** CITATIONS
SEE PROFILE

Fabrício Enembreck
Pontifícia Universidade Católica do Paraná (PUC-PR)
**143** PUBLICATIONS **1,418** CITATIONS
SEE PROFILE

# Analyzing the Impact of Feature Drifts in Streaming Learning

Jean Paul Barddal, Heitor Murilo Gomes and Fabrício Enembreck

Graduate Program in Informatics (PPGIa), Pontifícia Universidade Católica do Paraná, R. Imaculada Conceição, 1155

**Abstract.** Learning from data streams requires efficient algorithms capable of deriving a model accordingly to the arrival of new instances. Data streams are by definition unbounded sequences of data that are possibly non stationary, i.e. they may undergo changes in data distribution, phenomenon named concept drift. Concept drifts force streaming learning algorithms to detect and adapt to such changes in order to present feasible accuracy throughout time. Nonetheless, most of works presented in the literature do not account for a specific kind of drifts: feature drifts. Feature drifts occur whenever the relevance of an arbitrary attribute changes through time, also impacting the concept to be learned. In this paper we (i) verify the occurrence of feature drift in a publicly available dataset, (ii) present a synthetic data stream generator capable of performing feature drifts and (iii) analyze the impact of this type of drift in stream learning algorithms, enlightening that there is room and the need for dynamic feature selection strategies for data streams.

## 1 Introduction

Mining massives amount of data that arrive at rapid rates, namely data streams, is a recurring challenge. Extracting useful knowledge from these potentially unbounded sequences of data requires algorithms capable of acting within limited time, memory space and deal with its peculiarities i.e. concept drifts [9, 15] and evolutions [13]. Concept drifts occur when the data distribution changes over time and are divided in two types: real and virtual. Real concept drifts refer to changes in the conditional distribution of the target variable $y$ given the input (features) $\mathcal{D}$, while its distribution in the data input space $P[\boldsymbol{x}]$ may stay intact. Conversely, virtual concept drifts occur when the data distribution $P[\boldsymbol{x}]$ changes, independently of the conditional probability of the output values $P[y|\boldsymbol{x}]$ [8].

In this paper we review a specific kind of drift that is not commonly addressed in the literature: feature drifts. Feature drifts occur whenever the relevance of a feature (dimension) of a data stream grows or shrinks with time, enforcing the learning algorithm to adapt its model to ignore the irrelevant attributes and account for the newly relevant ones [14]. Several approaches on how to compute the relevance of a feature for the classification task were proposed in the literature, such as Entropy, Information Gain and Gini Index [10].

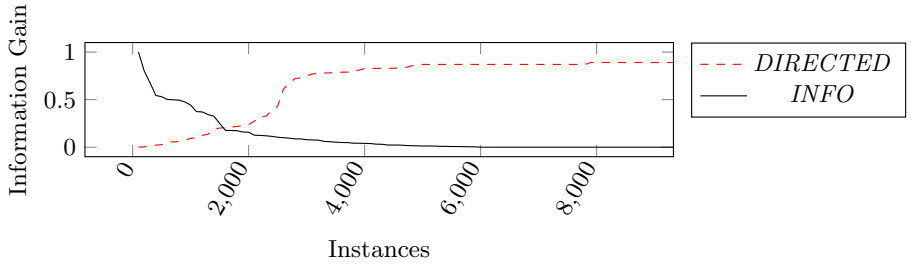



Fig. 1: Information Gain for two specific features of the Spam Corpus dataset.

In order to exemplify a feature drift, we refer to the e-mail spam detection system presented in [12]. This system was a result of a text mining process on an online news dissemination system. Essentially, this work intended on creating an incremental filtering of emails that classifies emails as spam or not and based on this classification, decides whether this email is relevant for dissemination among users. The dataset created contains 9,324 instances and 39,917 features, such that each attribute represents the presence of a single work (feature label) in the instance (e-mail). This dataset is known for containing a concept drift which occurs gradually around the instance of number 1,500 [1, 12].

In Fig. 1 we present a plot of the information gain [10] of two specific attributes presented in this problem, namely "directed" and "listinfo", where one can see that the the importance of these two attributes exchange gradually around instance 1,500.

This paper is divided as follows. The data stream learning and feature drift problems are specified in Section 2. In Section 3 we present a data stream generator able to simulate feature drifts. In Section 4 we empirically show the impact of feature drifts in two algorithms: an updatable naïve bayes algorithm and an incremental decision tree, namely Hoeffding Tree. Finally, in Section 5 we state the conclusions of this work and discuss envisioned future works.

## 2 Problem Statement

Let $\mathcal{S}$ be a data stream providing instances $i_t = (\boldsymbol{x}_t, y)$ intermittently, where $\boldsymbol{x}_t$ is a $d$-dimensional data object arriving at a timestamp $t$ and $y$ is its label. Instances $\boldsymbol{x}_i$ are labeled accordingly to values defined in $\mathcal{Y} = \{y_1, \ldots, y_c\}$. Also, let $\mathcal{D} = \{D_1, D_2, \ldots, D_d\}$ be the set features of a data stream where $d \geq 1$ is the dimensionality of the problem. It is assumed that $\mathcal{S}$ is unbounded, i.e. $|\mathcal{S}| \to \infty$, thus, it is not feasible to store all instances in memory before processing. This characteristic forces algorithms to either process data in limited size chunks or to incrementally process instances. Firstly, every instance $\boldsymbol{x}_i$ must be processed before an instance $\boldsymbol{x}_{i+1}$ becomes available, otherwise instances start to accumulate and the algorithm may have to discard them. Secondly, there is an inherent temporal aspect associated with a stream process, where the data distribution

may change over time, namely concept drift. Therefore, algorithms must also be able to detect and adapt to drifts, updating the algorithm's model.

**Definition 1.** *Let Eq. 1 denote a concept $C$, a set of prior probabilities of the classes and class-conditional probability density function [14]. Given a stream $\mathcal{S}$, instances $i_t$ retrieved will be generated by a concept $C_t$. If during each instant $t_i$ of $\mathcal{S}$ we have $C_{t_i} = C_{t_{i-1}}$, it occurs that the concept is stable. Otherwise, if between any two timestamps $t_i$ and $t_j$ occurs that $C_{t_i} \neq C_{t_j}$, we have a concept drift.*

$$C = \{(P[y_1], P[\boldsymbol{x}|y_1]), \ldots, (P[y_c], P[\boldsymbol{x}|y_c])\} \tag{1}$$

**Definition 2.** *Given a feature space $\mathcal{D}$ at a timestamp $t$, we are able to select the top discriminative subset $\mathcal{D}_t^* \subseteq \mathcal{D}$. A feature drift occurs if, at any two time instants $t_i$ and $t_j$, $\mathcal{D}_{t_i}^* \neq \mathcal{D}_{t_j}^*$ betides.*

In this paper we address the feature drift problem, where relevances of features of the data stream vary through time.

**Definition 3.** *Let $r(D_i, t_j) \in \{0, 1\}$ denote a function which determines the relevance of a feature $D_i$ in a timestamp $t_j$ of the stream. A positive relevance ($r(D_i, t_i) = 1$) states that $D_i \in \mathcal{D}^*$ in a timestamp $t_i$ and that it impacts the underlying probabilities $P[\boldsymbol{x}|y_i]$ of the concept $C_t$ in $\mathcal{S}$. A feature drift occurs whenever the relevance of an attribute $D_\alpha$ changes in a timespan between $t_j$ and $t_k$, as stated in Eq. 2.*

$$\exists t_j \exists t_k,\ t_j < t_k,\ r(D_\alpha, t_j) \neq r(D_\alpha, t_k) \tag{2}$$

Changes in $r(\cdot, \cdot)$ directly affect the ground-truth decision boundary to be learned by the inductive algorithm. Therefore, feature drifts can be seen as a specific type of real concept drift which can occur with or without changes in the data distribution $P[\boldsymbol{x}]$. As in other concept drifts, changes in $r(\cdot, \cdot)$ may occur during the stream, therefore enforcing algorithms to discard or adapt the model already learned, which is based on features that became irrelevant, which shall be replaced by the most relevant ones [14]. It is important to emphasize that feature drifts differ from concept drifts since concept drifts might occur without changes in attributes relevances but only in the a posteriori probabilities $P[\boldsymbol{x}|y]$.

Additionally, performing dynamic feature selection is desired since it provides a smaller subset of features that gives you as good or better accuracy in the predictive model, while requiring less data. Less attributes (dimensions) is desirable since it reduces the complexity of the model, leading to a smaller chance of overfitting and a model that is simple to understand and explain [5].

In the following section we present a data stream generator able to simulate feature drifts.

## 3 Simulating Feature Drifts

To verify the impact of feature drifts in existing streaming learning algorithms, we present a data stream generator that extends the SEA generator [16].

The generator here proposed simulates streams with $d > 2$ uniformly distributed features given by the user, where $\forall D_i \in \mathcal{D}, D_i \in [0; 10]$ and only two randomly picked features are relevant to the concept to be learned: $D_\omega$ and $D_\zeta$. As in [16], the class value $y$ is given accordingly to Eq. 3, where $\theta$ is a user-given threshold.

$$y = \begin{cases} 1, \ D_\omega + D_\zeta \leq \theta \\ 0, \ otherwise \end{cases} \tag{3}$$

Additionally, each instance synthesized has a 10% probability of being generated as noise.

To promote synthetic feature drifts in streams, we adopt the sigmoid framework stated in Eq. 4 and introduced in [4]. This model treats a feature drift as a combination of two pure distributions that characterizes concepts before and after the drift. The variables presented in Eq. 4 are the following: $f(t_i)$ is the probability that an instance $\boldsymbol{x}_i$ belongs to the prior concept, $1 - f(t_i)$ is the probability for the posterior concept, $w$ is the drift window size and $t_0$ is the drift moment.

$$f(t_i) = \frac{1}{(1 + e^{-w \times (t_i - t_0)})} \tag{4}$$

In [2] authors observe that Eq. 4 has a derivative at time $t_0$ equal to $f'(t_0) = s/4$ and that $\tan \ \alpha = f'(t_0)$, thus $\tan \ \alpha = s/4$. Also, $\tan \ \alpha = 1/W$ and as $s = 4 \ \tan \ \alpha$ then $\alpha = 4/W$, namely $t_0$ (time of drift), $w$ and $\alpha$ (phase angle). In this sigmoid model there are only two parameters to be specified: $t_0$ and $W$.

Nonetheless, it is important to emphasize that any decay function can be applied to simulate feature drifts.

## 4 Analysis

In this section we evaluate the accuracy of an incremental and updatable Naïve Bayes algorithm and an incremental decision tree, namely Hoeffding Tree [6], in both abrupt and gradual feature drifts. Firstly, we briefly introduce the evaluated algorithms and the experimental protocol adopted. Finally, we discuss the results obtained.

### 4.1 Evaluated Algorithms

**Updatable Naïve Bayes** The updatable Naïve Bayes algorithm is an incremental version of the popular Naïve Bayes algorithm. Both algorithms rely on

the assumption that all attributes of the dataset are independent, with the exception of the output value $y$, which depends on all others $D_1, \ldots, D_d$. Therefore, these algorithms compute the output value for an input instance $\boldsymbol{x}_i$ as stated in Eq. 5, determining the value of $y$ that maximizes the probability $P[\boldsymbol{x}_i|y]$.

$$P[\boldsymbol{x}_i|y] = \frac{P[y|\boldsymbol{x}_i] \times P[\boldsymbol{x}_i]}{P[y]} \tag{5}$$

In order to compute probabilities in a streaming environment, the Updatable Naïve Bayes stores a contingency table, therefore, no windowing process is needed whatsoever.

**Hoeffding Tree** Hoeffding Trees algorithms construct decision trees by using constant memory and constant time per sample [6]. These trees are built by recursively replacing leaves with decision nodes, as data arrives. Different heuristic evaluation functions are used to determine whether a split should be performed or not, such as Gain Ratio, Entropy and Gini Coefficient [10]. To do so, Hoeffding Trees assume that the input data meets the Hoeffding bound [11].

Assuming a random variable $r \in \mathbf{R}$ with range $R$, a number of independent observations $n$, the mean computed by the latter observations $\bar{n}$; the Hoeffding Inequality states that with probability $1 - \delta$ the true mean of a variable is at least $\bar{r} - \epsilon$, where $\epsilon$ is given by Eq. 6 and $\delta$ is a user-given confidence bound.

$$\epsilon = \sqrt{\frac{R^2 \ln\left(\frac{1}{\delta}\right)}{2n}} \tag{6}$$

The Hoeffding bound is able to give results regardless the probability distribution that generates data. However, the number of observations needed to reach certain values of $\delta$ and $\epsilon$ are different across different probability distributions [3]. Generally, with probability $1 - \delta$, one can say that one attribute is superior when compared to others when observed difference of information gain (or any other metric that computes the importance of an attribute) is greater then $\epsilon$.

Finally, all tree's nodes maintain statistics about the data used to derive itself. Periodically, Hoeffdings Trees discard nodes of the tree that are not accessed during traverses and replaces them by new ones accordingly to the Hoeffding bound and the chosen split function.

### 4.2 Experimental Protocol

Five different scenarios are evaluated in this section. The first scenario is the Spam Corpus dataset presented in [12], while the other four adopt the generator presented in Section 3 and were parametrized as follows:

- FD-1: 50,000 instances, $\theta = 7$ and $d = 10$
  - Drift 1: $t_0 = 25,000$, $w = 1$;
- FD-2: 50,000 instances, $\theta = 7$ and $d = 10$
  - Drift 1: $t_0 = 25,000$, $w = 1,000$;

- FD-3: 100,000 instances, $\theta = 9.5$ and $d = 10$
  - Drift 1: $t_0 = 34,000$, $w = 1$;
  - Drift 2: $t_0 = 67,000$, $w = 1$;
- FD-4: 100,000 instances, $\theta = 9.5$ and $d = 10$
  - Drift 1: $t_0 = 34,000$, $w = 1,000$;
  - Drift 2: $t_0 = 67,000$, $w = 1,000$;

In our experiments accuracy is measured using the Prequential test-then-train method. We adopted the Prequential procedure [7] due the the monitoring of the evolution of performance of models over time although it may be pessimistic in comparison to the holdout estimative. Nevertheless, authors in [7] observe that the prequential error converges to an periodic holdout estimative when estimated over a sliding window. Along these lines, we determined an evaluation sliding window of 1,000 instances for these experiments.

Finally, all experiments here presented were implemented and evaluated under the Massive Online Analysis (MOA) framework [4].

### 4.3 Results Obtained

In Fig. 2a one can see that accuracy drops by 60% during the known feature drift and slowly recovers after approximately 3,500 instances.

In Figs. 2b through 2e we present the results obtained by the Naïve Bayes and the Hoeffding Tree algorithms in the FD-1, FD-2, FD-3 and FD-4 experiments, respectively.

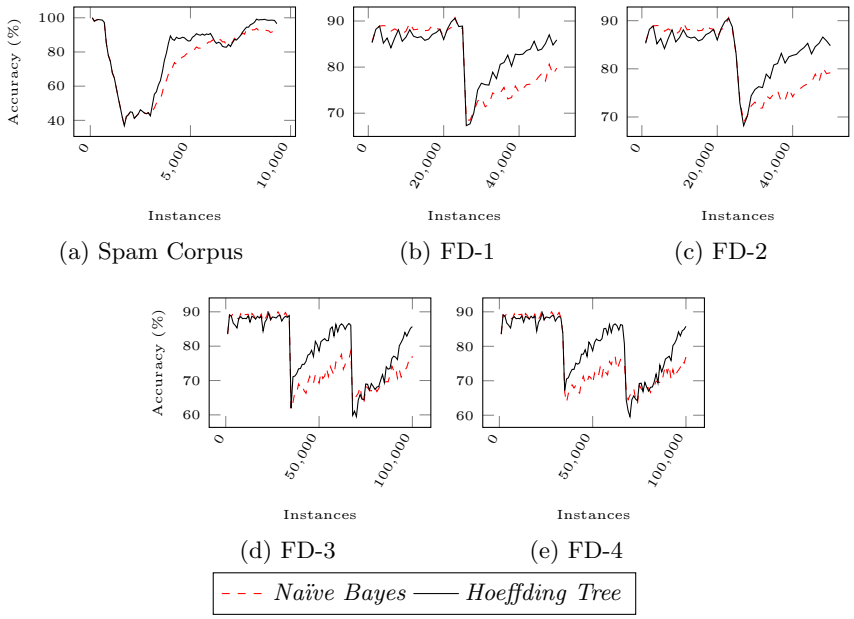
In Figs. 2b and 2c one can see the impact of one feature drift during the stream. In both cases, it is, abrupt and gradual changes, both algorithms has its accuracy damped in 20% and the Naïve Bayes fails to completely recover until the end of the stream.

Additionally, in Figs. 2d and 2e one can see that impact of two drifts in accuracy for both algorithms. Again, the mean accuracy drops by 30%, showing the difficulty of adapting to both abrupt and gradual feature drifts.

The results here presented enable us to argue that existing algorithms do not account for the possibility of feature drifts. Even Hoeffding Trees, which perform feature selection during the stream, fail to quickly adapt to changes in features' relevances, showing that there is room and the need for dynamic feature selection algorithms for data streams.

## 5 Conclusion

In this paper we analyzed the feature drift problem. Feature drifts differ from conventional concept drifts since they do not occur accordingly to changes in the data distribution, but on the relevance of each attribute in the concept to be learned. Additionally, we presented a data generator capable of synthesizing data streams with this peculiarity. Finally, we benchmarked an incremental and updatable Naïve Bayes classifier and an incremental decision tree on synthetic




(a) Spam Corpus (b) FD-1 (c) FD-2

(d) FD-3 (e) FD-4

Fig. 2: Accuracy obtained during experiments with feature drifts.

data streams with feature drifts, showing the impact of feature drifts in their accuracy. We must emphasize that even the Hoeffding Tree fails to quickly adapt to feature drifts, an important trait since it possesses an embedded feature selection algorithm to determine splits in real-time processing, which is however, performed accordingly to user-given parameters and not automatically.

The results here presented highlight the inefficiency of algorithms on tracking which attributes are relevant for classification in data streams. Therefore, dynamic feature selection algorithms are of utmost importance to quickly detect and adapt to feature drifts.

In future works we plan to verify the efficiency of state-of-the-art algorithms with the addition of feature selection algorithms using periodical verifications of feature relevances accordingly to a landmark windowing technique. Furthermore, we plan to study the impact of feature evolutions, i.e. appearance and disappearance of features, in streaming learning environments.

## References

1. Jean Paul Barddal, Heitor Murilo Gomes, and Fabrício Enembreck. Sfnclassifier: A scale-free social network method to handle concept drift. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing (SAC)*, SAC 2014. ACM, March 2014.

2. A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà. New ensemble methods for evolving data streams. In *Proc. of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 139–148. ACM SIGKDD, Jun. 2009.
3. Albert Bifet and Ricard Gavaldà. Learning from time-changing data with adaptive windowing. In *In SIAM International Conference on Data Mining*, 2007.
4. Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. MOA: Massive online analysis. *The Journal of Machine Learning Research*, 11:1601–1604, 2010.
5. Vitor R. Carvalho and William W. Cohen. Single-pass online learning: Performance, voting schemes and online feature selection. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 548–553, New York, NY, USA, 2006. ACM.
6. Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 71–80, New York, NY, USA, 2000. ACM.
7. J. Gama and P. Rodrigues. Issues in evaluation of stream learning algorithms. In *Proc. of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 329–338. ACM SIGKDD, Jun. 2009.
8. João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4):44:1–44:37, March 2014.
9. Joao Gama. *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC, 1st edition, 2010.
10. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.
11. Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.
12. Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. Dynamic feature space and incremental feature selection for the classification of textual data streams. In *in ECML/PKDD-2006 International Workshop on Knowledge Discovery from Data Streams. 2006*, page 107. Springer Verlag, 2006.
13. Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, and Bhavani M. Thuraisingham. Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Trans. Knowl. Data Eng.*, 23(6):859–874, 2011.
14. Hai-Long Nguyen, Yew-Kwong Woon, Wee-Keong Ng, and Li Wan. Heterogeneous ensemble for feature drifts in data streams. In Pang-Ning Tan, Sanjay Chawla, ChinKuan Ho, and James Bailey, editors, *Advances in Knowledge Discovery and Data Mining*, volume 7302 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg, 2012.
15. Jonathan A. Silva, Elaine R. Faria, Rodrigo C. Barros, Eduardo R. Hruschka, André C. P. L. F. de Carvalho, and João Gama. Data stream clustering: A survey. *ACM Comput. Surv.*, 46(1):13:1–13:31, July 2013.
16. W. Nick Street and Y. Kim. A streaming ensemble algorithm (sea) for large-classification. In *Proc. of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 377–382. ACM SIGKDD, Aug. 2001.