

Chapter 2

Learning in Dynamic Environments

2.1 Introduction

The volume of data is rapidly increasing due to the development of the technology of information and communication. This data comes in the form of streams. Learning from this ever-growing amount of data requires a continuous learning over time. Traditional one-shot memory-based learning methods trained offline from a fixed size of historic data set are not adapted to learn from these data streams. This is because firstly, it is not feasible to register all the data samples over time, and secondly the generated models become quickly obsolete due to the occurrence of changes, also known as “concept drift,” in their environments.

Therefore, as we have seen in Chap. 1, online self-adaptive learning scheme is required in order to accommodate the new information carried out by the new incoming data samples and to unlearn the obsolete or outdated ones due to the changes in the learner’s or model’s environments.

In this chapter, the problem of drifting data streams in dynamic environments is formalized, and its framework is defined. Then, the kinds and characteristics of the concept drift are presented. Finally, the real-world applications generating drifting data streams are discussed. The goal is to give a picture of the problem of learning from data streams in dynamic environments, its causes, sources, and characteristics in order to discuss later alternatives to solve this problem.

2.2 Concept Drift Framework

2.2.1 Incremental Learning

Let X_L be a set of historic labeled data samples available at time t . It contains n labeled samples: (x_i, y) , $i = 1, \dots, n, y \in \{1, \dots, c\}$, collected in the past until the time t and distributed into c classes. Let L_t be the learner (e.g., classifier) built using X_L . L_t is used to predict the class label y_{t+1} for a new unseen data sample x_{t+1} . Since after this classification the label y_{t+1} is available, then the learner L_t can be updated by integrating (x_{t+1}, y_{t+1}) to its learning or training set. If the update of L_t is achieved by using only the new labeled data sample, and not the whole X_L , then the learning is called incremental learning [10–12].

As we have seen in Chap. 1, a concept or a source S can be defined as the classes' joint probabilities $\{P(y_1, x), P(y_2, x), \dots, P(y_c, x)\}$ which in their turn are defined by the classes' prior probabilities $\{P(y_1), P(y_2), \dots, P(y_c)\}$ and the classes' conditional probabilities $\{P(y_1|x), P(y_2|x), \dots, P(y_c|x)\}$ (see (1.2)):

$$S = \{(P(y_1), P(x|y_1)), (P(y_2), P(x|y_2)), \dots, (P(y_c), P(x|y_c))\} \quad (2.1)$$

Every data sample x_i is generated by the source S_i . Therefore, if $S_1 = S_2 = \dots = S_n = S$, then the concept, defined by S or D , is stable. The incremental model can improve its performance by approximating D as the number of incoming data increases. Thus, the hypothesis (*concept*) learned before are still valid and can be efficiently approximated whenever the number of data samples increases to infinite. This is known by stable or stationary concept.

Consequently, incremental learning [10–12] is a suitable online learning scheme allowing to learn from infinite streams of data samples using limited time and memory size. Therefore, incremental learning methods do not require the availability of an initial complete training set since they continue to learn from the incoming data samples over time. However, they assume that the hypotheses (source, concept, distribution, etc.) learned before are always valid for the new incoming data. This reduces the ability of incremental learning methods to evolve the model (predictor, classifier) at the same rate as data streams.

In this context, they are considered to be suitable for learning from data streams since there is no control on their order of arrive nor their representativeness (i.e., the data samples may not be independent, and they may not be randomly generated from the source S). Moreover, they are theoretically able to continuously improve their performance as the number of incoming data increases. However, this can be true *if and only if* we assume that the hypotheses (S or D) generating the data streams are the same over time.

Example 2.1: Comparison Between Static and Incremental Classifiers Let us consider two classes described in one-dimensional feature space as follows:

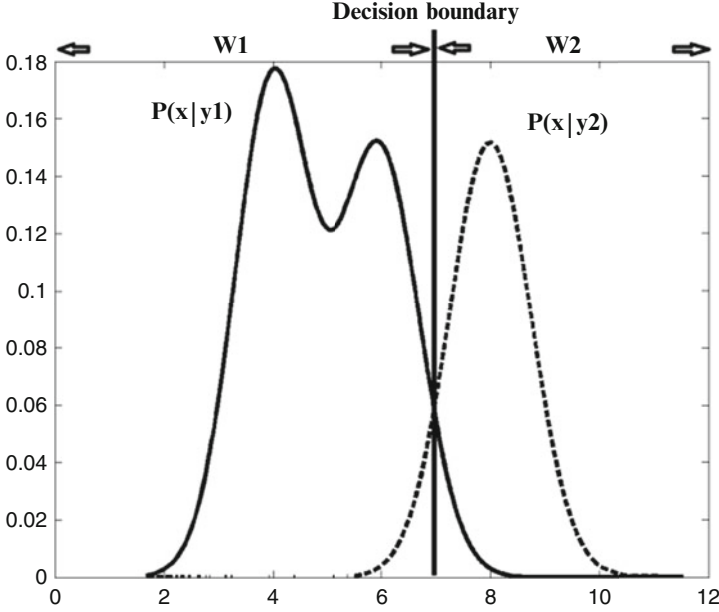


Fig. 2.1 Conditional probability densities for Gaussian classes W_1 and W_2 and the corresponding optimal boundary decision learned from a complete learning set

$$\begin{aligned} x \in W_1 &\Rightarrow x \sim N(M_1 = 4, \Sigma_1 = 1) + N(M_1 = 6, \Sigma_1 = 1) \\ x \in W_2 &\Rightarrow x \sim N(M_2 = 8, \Sigma_2 = 1) \end{aligned}$$

We suppose that both classes have the same prior probability: $P(y_1) = P(y_2)$. We can see that the data samples of W_1 belong to two different Gaussian distributions. Figure 2.1 presents the conditional probabilities $P(x|y_1)$ and $P(x|y_2)$ according to classes W_1 and W_2 as well as the optimal boundary decision. Let us suppose that the available data samples belonging to W_1 in X_L were generated only from $x \sim N(M_1 = 4, \Sigma_1 = 1)$. Therefore, the learner will learn the conditional probabilities $P(x|y_1)$ and $P(x|y_2)$ depicted in Fig. 2.2. If the classifier is static, the new incoming data samples will be classified according to the decision boundary depicted in Fig. 2.2. Based on (1.7), the patterns between the expected decision boundary of Fig. 2.1 and the one learned by the static classifier will be misclassified since they will be classified in W_2 while they are in W_1 . An incremental classifier will update its decision boundary according to the incoming new data samples. The ones of the latter generated by $x \sim N(M_1 = 6, \Sigma_1 = 1)$ will allow the incremental classifier to improve its performance (accuracy or classification rate) by updating its decision boundary (see Fig. 2.1).

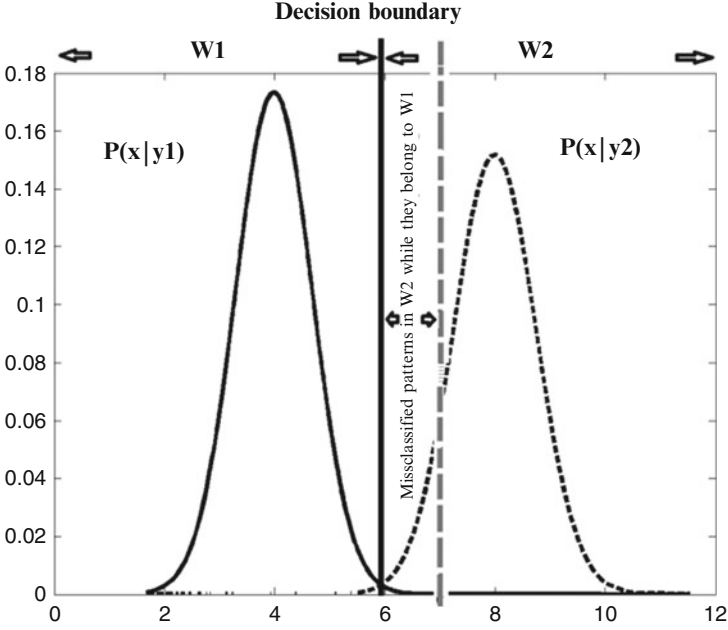


Fig. 2.2 Conditional probability densities for Gaussian classes W_1 and W_2 and the corresponding optimal decision boundary learned from an incomplete learning set. The *gray dashed* decision boundary represents the boundary decision in the case of complete learning set. All the data samples between these two decision boundaries will be misclassified in W_2 while they belong to W_1

2.2.2 Adaptive Learning

One major challenge arises when the underlying source generating the data is not stationary: $S_1 \neq S_2 \neq \dots \neq S_n$. This leads to a change in the data distribution according to a single feature, to a combination of features or in the class boundaries. This is known as concept drift. In this case, the assumption of data identically distributed is no more satisfied, and the incremental learning is no longer able to approximate the distribution of the new incoming data samples. Indeed, incremental learning considers the already learned concepts are valid. This is the case in many real-world domains where the concept of interest may depend on some hidden context, not given explicitly in the form of predictive features.

Example 2.2: Comparison Between Static, Incremental, and Adaptive Classifiers Let us consider two Gaussian classes described in one-dimensional feature space as follows:

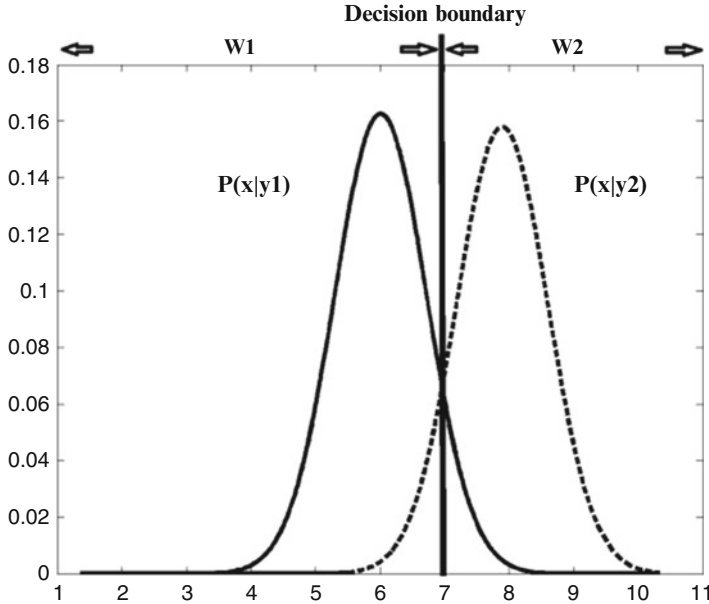


Fig. 2.3 Conditional probability densities $P(x|y_1)$ and $P(x|y_2)$ for Gaussian classes W_1 and W_2 and the corresponding optimal decision boundary learned before the concept drift in W_1

$$\begin{aligned} x \in W_1 &\Rightarrow x \sim N(M_2 = 6, \Sigma_2 = 1) \\ x \in W_2 &\Rightarrow x \sim N(M_1 = 8, \Sigma_1 = 1) \end{aligned}$$

We suppose that both classes have the same prior probability: $P(y_1) = P(y_2)$. Figure 2.3 presents the conditional probabilities $P(x|y_1)$ and $P(x|y_2)$ according to classes W_1 and W_2 as well as their optimal boundary decision. Let us now suppose that the new incoming data samples do not follow any more $x \sim N(M_1 = 6, \Sigma_1 = 1)$ but are generated by new source or distribution defined by $x \sim N(M_1 = 4, \Sigma_1 = 1)$. Therefore, W_1 will be defined by the new conditional probability density depicted in Fig. 2.4. Due to this concept drift in W_1 ($x \sim N(M_1 = 6, \Sigma_1 = 1) \rightarrow x \sim N(M_1 = 4, \Sigma_1 = 1)$), the decision boundary must be updated as it is depicted in Fig. 2.4. However, the incremental learning cannot update correctly the boundary decision since it considers all the data samples, and therefore their sources or distributions, are valid (see Fig. 2.5).

Consequently, the data samples generated by the old distribution or source of W_1 , $x \sim N(M_1 = 6, \Sigma_1 = 1)$, must be removed or unlearned, and only the data samples generated by the new source or distribution ($x \sim N(M_1 = 4, \Sigma_1 = 1)$) must be used to update the decision boundary.

In order to achieve the most accurate classification or prediction in the presence of concept drift, the learner should be able to track such drift and quickly adapt to it. Therefore, self-adaptive online learning [13–16] is the most adequate learning

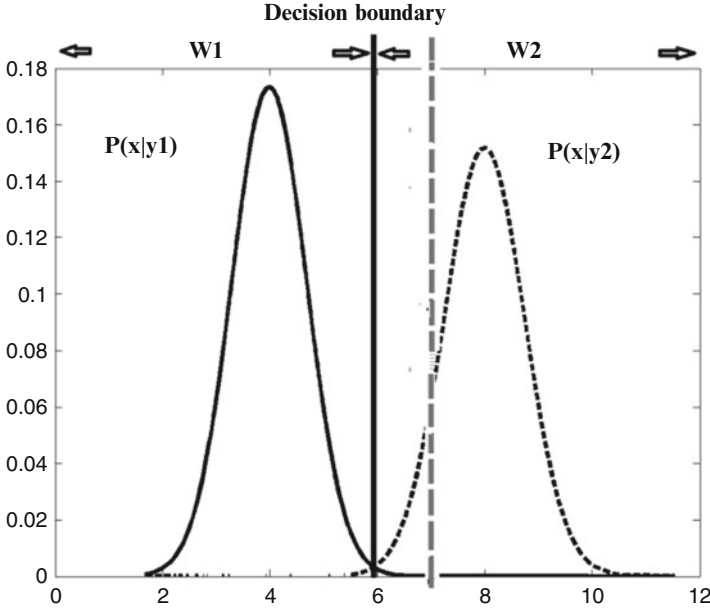


Fig. 2.4 Conditional probability densities $P(x|y_1)$ and $P(x|y_2)$ for Gaussian classes W_1 and W_2 and the corresponding optimal decision boundary learned using the data samples generated by the new source of W_2 and removing or unlearning the ones generated by the old source. The *gray dashed* boundary decision represents the initial boundary decision before the concept drift in W_1

scheme to learn from evolving data streams in dynamic environments since they integrate a forgetting mechanism of outdated or obsolete data. To achieve that, two questions must be answered: (1) how to track concept drift and (2) how to adapt the learner parameters and structure in order to react to this concept drift. To answer these questions, the drift causes, types, and characteristics will be detailed in the next. This will allow defining how data samples that are representative of the new concept (the new source or distribution generating the data) can be determined and how they can be used to adapt the classifier parameters and structure.

2.3 Causes and Kinds of a Concept Drift

Based on (1.4) and (1.7), there are FOUR terms that are used by the Bayes formula to achieve the classification task:

- The prior probability $P(y_i)$ of each class W_i
- Its conditional probability $P(x|y_i)$
- The posterior probability $P(y_i|x)$
- The marginal or prior probability $P(x)$ of x

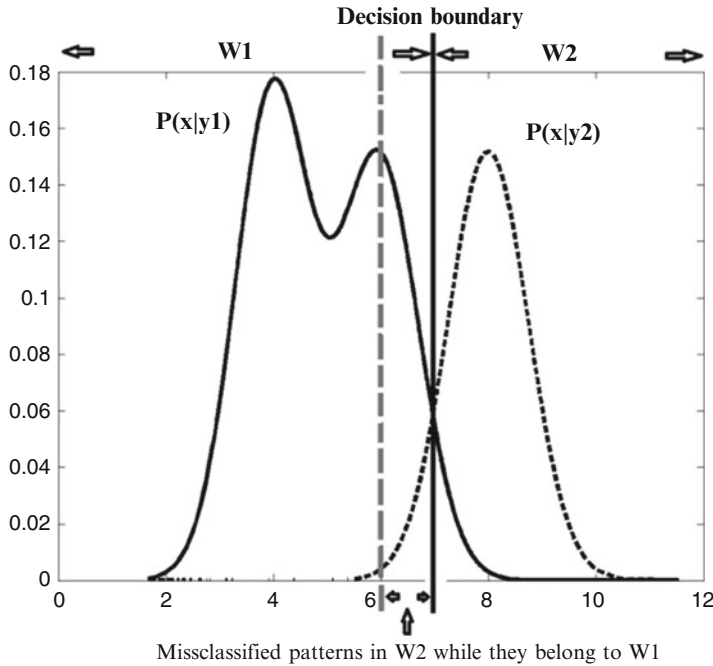


Fig. 2.5 Conditional probability densities $P(x|y_1)$ and $P(x|y_2)$ for Gaussian classes W_1 and W_2 and the corresponding optimal decision boundary learned using all the data samples generated by the old and new concepts of W_1 . The *gray dashed line* represents the decision boundary learned by using only the data samples generated from the new source. All the data samples between these two decision boundaries will be misclassified in W_2 while they belong to W_1

$P(x)$ is constant for all the classes since it acts as a normalization or evidence factor. Therefore, a concept drift may result due to a change in:

- The prior probability $P(y_i)$ of a class W_i
- The conditional probability $P(x|y_i)$ in a class W_i
- The posterior probability $P(y_i|x)$ of a class W_i
- A combination of them

It is worth mentioning that a change in the class prior probability leads to class imbalance, novel class emergence, existing classes' fusion, or existing classes' splitting.

These changes can cause two kinds of concept drift: real and virtual. The real concept drift refers to changes in the classes' posterior probabilities $P(y_i|x)$. This means that the target concept, y , for a pattern x with the same values of features will change in response to the occurrence of a drift. Therefore, this kind of drift directly impacts the decision boundary, which in turn decreases the performance of the learner if the latter is not updated. The virtual concept drift refers to changes in the classes' conditional probabilities $P(x|y_i)$ without impacting the posterior

probabilities $P(y_i|x)$. Therefore, this kind of concept drift impacts the data distribution within the same class in the feature space without affecting the corresponding decision boundaries.

Example 2.3: Real drift caused by a change in the classes' prior probabilities Let us take the case of two Gaussian classes described in one-dimensional space as follows:

$$\begin{aligned} x \in W_1 &\Rightarrow x \sim N(M_1 = 2, \Sigma_1 = 1) \\ x \in W_2 &\Rightarrow x \sim N(M_2 = 5, \Sigma_2 = 1) \end{aligned}$$

Let us suppose that W_1 represents the healthy human subjects, while W_2 gathers the ill ones. Firstly, we suppose that the learning set X_L includes 200 data samples: 160 data samples belong to class W_1 , and the other 40 ones belong to class W_2 . Therefore, the prior probabilities for W_1 and W_2 are: $P(y_1) = 0.8$ and $P(y_2) = 0.2$. $P(y_1)$ is significantly bigger than $P(y_2)$ because in normal conditions, we expect more healthy human subjects than ill ones.

Let us suppose that within the next 800 inspected human subjects: 40 were assigned to W_1 , while 760 ones assigned to W_2 . This means that $P(y_1)$ will be significantly decreased to 0.2, while $P(y_2)$ will be increased to 0.8. This significant change in the prior probabilities of both classes indicates a change (drift) in the classifier environments. This change can be due to an epidemic spread. The latter increases significantly the number of contaminated human subjects (e.g., W_2) and reduces the number of healthy human subjects (e.g., W_1) according to the total number of inspected human subjects. This change entails a change in the posterior probabilities of both classes as well as in their optimal decision boundary as we can see in Fig. 2.6. If the original decision boundary is not updated in response to this change, all the patterns located between the original and new decision boundaries will be misclassified. This leads to decrease the classifier performance (the rate of patterns correctly classified).

We must mention here that the conditional probabilities $P(x|y_1)$ and $P(x|y_2)$ of classes W_1 and W_2 did not change; they are the same before and after the change in the classes' prior probabilities.

Example 2.4: Real Concept Drift Caused by a Change in the Classes' Posterior Probabilities Let us take the example of the classifier of the electricity price tendency, treated in Examples 1.2 and 1.8. Let us suppose that the prices of oil have been changed (decreased or increased) because of a political or economic crisis event. Then for the same values of the input vector $x = (x^1, x^2, x^3)$, representing the electricity demand in the studied area, the electricity demand in the adjacent area, and the scheduled electricity transfer between these two areas, the estimated class (up/down) may become different from the one estimated without the occurrence of this unpredicted event. If this change or drift is not taken into account, the classifier prediction accuracy will be decreased.

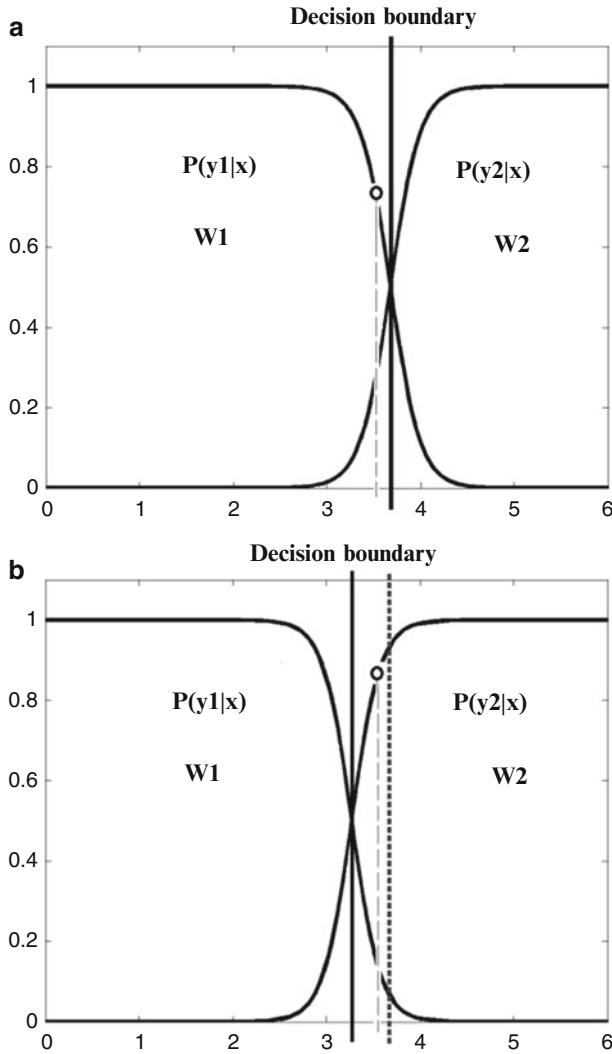


Fig. 2.6 Posterior probabilities $P(y_1|x)$ and $P(y_2|x)$ for the two Gaussian classes W_1 and W_2 as well as their corresponding decision boundary before (a) and after (b) the change in the classes' prior probabilities $P(y_1)$ and $P(y_2)$. The *dashed gray line* in (b) shows the decision boundary before the change in the prior probabilities $P(y_1)$ and $P(y_2)$ of W_1 and W_2 . For the round point representing a data sample, it belongs to W_1 before the drift (a), and after the drift it belongs to W_2 (b). We can see clearly that the decision boundary and posterior probabilities have been changed in response to this change in the prior probabilities of both classes

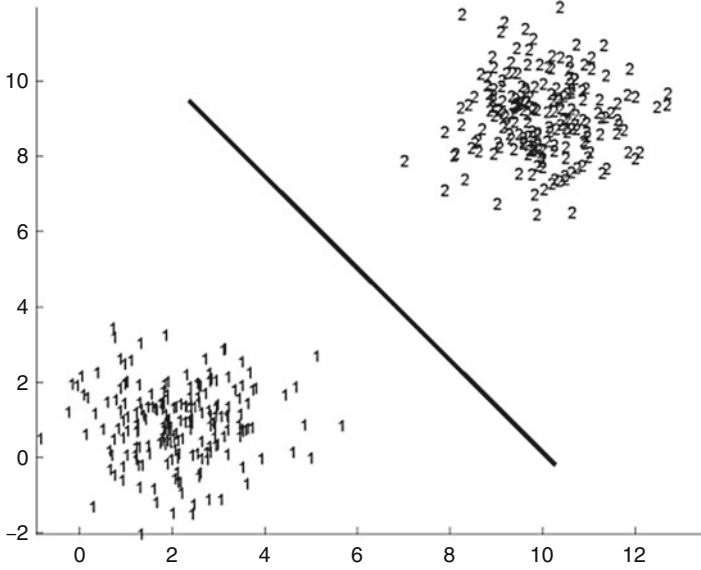


Fig. 2.7 Two Gaussian classes and their corresponding decision boundary in two-dimensional feature space

Example 2.5: Virtual Concept Drift Caused by a Change in the Classes' Conditional Probabilities Let us take the case of two Gaussian classes described in two-dimensional space as follows:

$$\begin{aligned} x \in W_1 &\Rightarrow x \sim N(M_1 = (2, 1), \Sigma_1 = (1, 1)) \\ x \in W_2 &\Rightarrow x \sim N(M_2 = (10, 9), \Sigma_2 = (1, 1)) \end{aligned}$$

Figure 2.7 depicts these two classes in the feature space as well as their corresponding decision boundary. Let us suppose that a drift has occurred in the class W_1 leading to a change in its mean values as follows:

$$x \in W_1 \Rightarrow x \sim N(M_1 = (3, 3), \Sigma_1 = (1, 1))$$

This change in the parameters (mean values) of the normal law generating the data samples is represented as a change (a move) of the location of W_1 in the feature space (see Fig. 2.8). This drift changes the data spatial distribution of W_1 in the feature space without impacting the decision boundary. Indeed, the performance of the classifier with the initial decision boundary (see Fig. 2.7) will not be impacted by this drift in W_1 since both classes after the drift remain perfectly separated by the initial decision boundary (see Fig. 2.8). Therefore, the drift in the class W_1 is virtual since it does not impact the classifier performance.

Let W_1 and W_2 be, respectively, the normal and failure operation conditions of a machine. The classifier aims at assigning an incoming pattern representing the

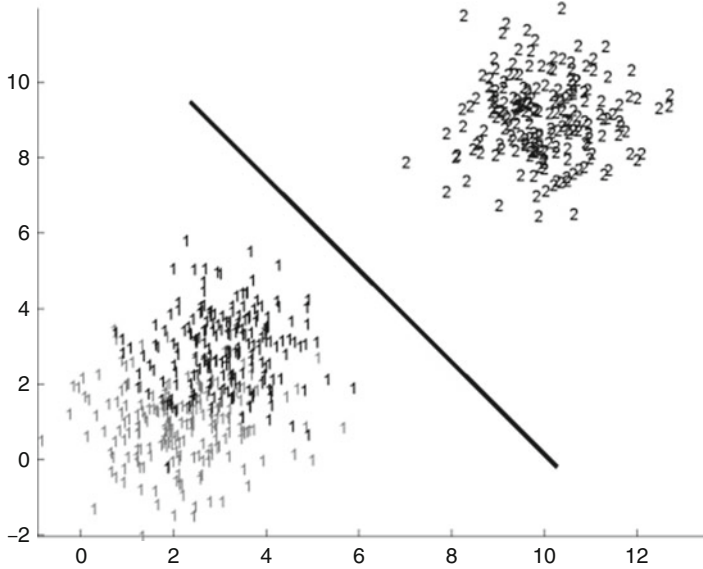


Fig. 2.8 Virtual drift in the class W_1 leading to change its spatial location or distribution in the feature space. The *gray points* of W_1 indicate the initial patterns belonging to W_1 before the drift. This drift does not impact the decision boundary since the initial one still able to separate perfectly both classes after the drift in W_1

machine's current operation conditions into normal or failure classes. When this machine starts to malfunction, its performance to accomplish a task decreases over time. However, as long as this performance remains acceptable (greater than the threshold defined for the failure representing unacceptable decrease in the machine's performance), the classifier continues to classify properly the incoming patterns as belonging to the class of normal operation conditions although the characteristics of the normal class are drifting over time. Therefore, in this case, the concept drift is virtual.

Example 2.6: Virtual Concept Drift Becoming Real Concept Drift Let us take Example 2.5 where the conditional probability $P(x|y_1)$ of class W_1 changed as follows:

Before drift: $x \in W_1 \Rightarrow x \sim N(M_1 = (2, 1), \Sigma_1 = (1, 1))$.

After drift: $x \in W_1 \Rightarrow x \sim N(M_1 = (3, 3), \Sigma_1 = (1, 1))$.

Let us suppose now that a new concept drift occurred in the conditional probability $P(x|y_1)$ of W_1 as follows (see Fig. 2.9):

$$x \in W_1 \Rightarrow x \sim N(M_1 = (5, 6), \Sigma_1 = (1, 1))$$

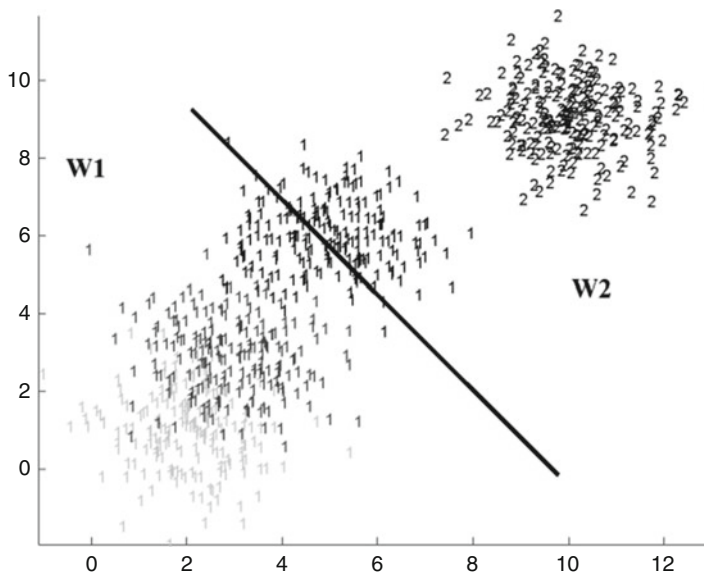


Fig. 2.9 Drift in the class W_1 passing from virtual to real concept drift. When the drift in W_1 becomes real, the decision boundary is adversary impacted since the patterns of W_1 enter the area of W_2 in the feature space. All the patterns of W_1 located in the area of W_2 in the feature space will be then misclassified

Some of the data samples belonging to the new concept drift of W_1 will occupy the zone of class W_2 in the feature space. Therefore, these data samples will be misclassified by the static classifier since the latter will assign them to W_2 while they belong to W_1 . Consequently, the classifier performance (accuracy) will be decreased if this drift is not taken into account in order to update the decision boundary of the classifier.

Let us suppose that W_1 represents the class of spam while W_2 is the class gathering the legitimate emails. If the spammer looks to trick the spam filter (the classifier) by changing its behavior in order to be similar to the one of legitimate emails (class W_2), then the spatial position of W_1 in the feature space will be moved in order to occupy the zone of W_2 . When the patterns of W_1 enter the zone in the feature space that is considered by the classifier (spam filter) as the behavior of legitimate mails (W_2), then the spam emails will be considered as legitimate. This will lead to decrease the performance of the spam filter. Therefore, the decision boundary of the classifier (spam filter) must be updated in order to take into account the change in the behavior of the spam emails induced by the spammer to trick the spam filter. Figure 2.10 shows the new decision boundary that allows maintaining the classifier performance for the example of Fig. 2.9.

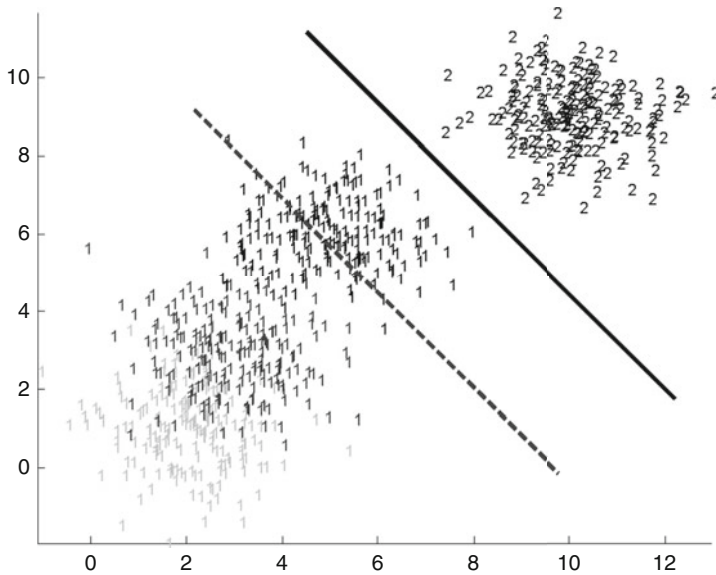


Fig. 2.10 Updating the decision boundary of the classifier in order to maintain its performance when the drift in the class W_1 passed from virtual to real concept drift. The *solid line* is the updated decision boundary of the initial one (in *gray dashed line*) before the drift

2.4 Concept Drift Description

A drift can be represented according to different criteria which are used to describe how the new concept replaces the old one [17–19]. These criteria give indications about the drift period, its speed, its intensity or severity, its frequency, and whether it can be detected or not. These characteristics are very useful to guide the choice and to define the framework of the methods and tools suitable to handle concept drift.

2.4.1 Drift Speed

The drift duration, called also drifting time or drift width, is the number of time steps for a new concept to replace the old one in the sense that no data samples of the old samples will occur. According to [18], speed is the inverse of the drifting time in the sense that a higher speed is related to a lower number of time steps and a lower speed is related to a higher number of time steps. Therefore, the drift speed V_d is calculated by

$$V_d = \frac{1}{t_{de} - t_{ds}} \quad (2.2)$$

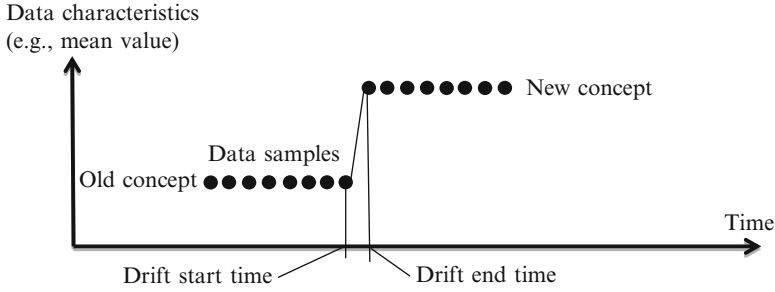


Fig. 2.11 Abrupt drift

where t_{de} and t_{ds} are, respectively, the time when the drift ends and the time when the drift starts.

According to its speed, a drift can be either abrupt or gradual:

- Abrupt drift occurs when the new concept suddenly replaces the old one in short drifting time. This kind of drift immediately deteriorates the learner performance, as the new concept quickly substitutes the old one (see Fig. 2.11).
- Gradual drift occurs when the drifting time is relatively large. This kind of drift is hard to detect since it creates a period of uncertainty due to the cohabitation of both old and new concepts. Gradual drift can be either probabilistic or continuous:
 - Gradual probabilistic drift refers to a period when both new concept generated by the source S_2 and old concept generated by the source S_1 cohabit. In other words, there is a weighted combination between data samples generated by S_1 (old concept) and the ones generated by S_2 (new concept). As time passes, the probability of sampling from S_1 decreases, whereas the probability of sampling from S_2 increases until the new concept totally replaces the old one (see Fig. 2.12).
 - Gradual continuous or incremental drift corresponds to the case where the concept itself continuously changes from the old to the new concept, by suffering small modifications at every time step. Therefore, during the continuous or incremental change, the new concept does not yet appear; the patterns representing the continuous drift do not have the same characteristics. It is worth to mention that these changes are so small that they are only noticed during a long time period (see Fig. 2.13). When the continuous drift is ended, then the new concept appears, and starting from this time, the incoming patterns are generated from the same source.

Example 2.7: Probabilistic Gradual Drift Let us suppose that the incoming data samples arrive within batches. Each batch contains 100 data samples. Let us suppose that the patterns or data samples in the first batch are generated by the source S_1 . Then let us suppose that in the second batch, 80 patterns were generated by the source S_1 (old concept), while 20 patterns were generated by the new source S_2 (new concept). In the

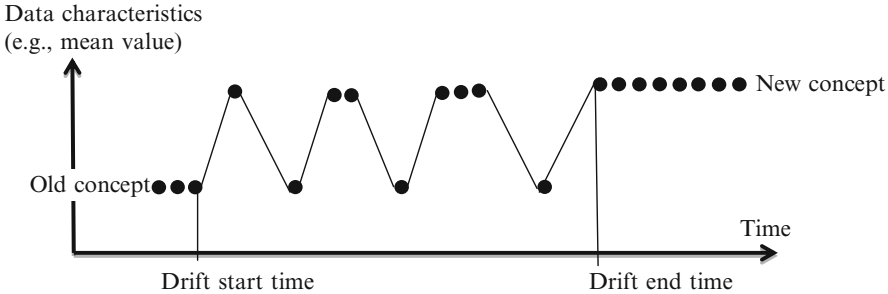


Fig. 2.12 Gradual probabilistic drift

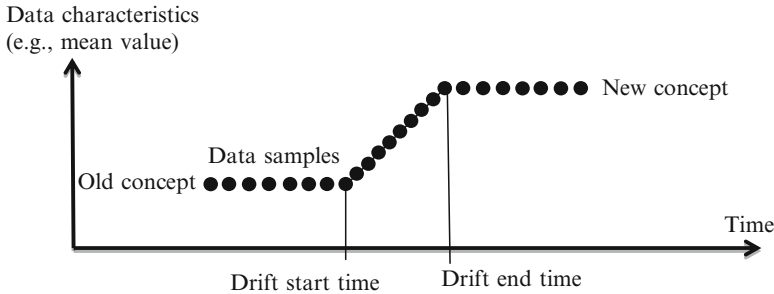


Fig. 2.13 Gradual continuous drift

third batch, 50 patterns are generated by the S_1 , and the other 50 patterns are generated by the new source S_2 . Finally, in the fourth and fifth batches, all the patterns are generated by the new source S_2 . Therefore, this drift replaces gradually the old concept (generated by S_1) by the new one (generated by S_2).

Figure 2.14 shows the probabilities of occurrence of patterns belonging to the old and new concepts. We can see that the probability P_1 of occurrence of patterns generated by S_1 decreases, while the one P_2 for patterns generated by S_2 increases over time. At the end, P_1 will be equal to 0, while P_2 is equal to 1 in order to indicate that the new concept has replaced completely the old one.

2.4.2 Drift Severity

The severity refers to the amount of changes caused by the drift occurrence. The drift severity can be high or low (partial). A high or global severity (see Fig. 2.15b) means that the old concept has been completely changed. Therefore, the whole region occupied by this old concept will be impacted by the drift. A low or partial severity (see Fig. 2.15c) refers to a change impacting only a part of the region

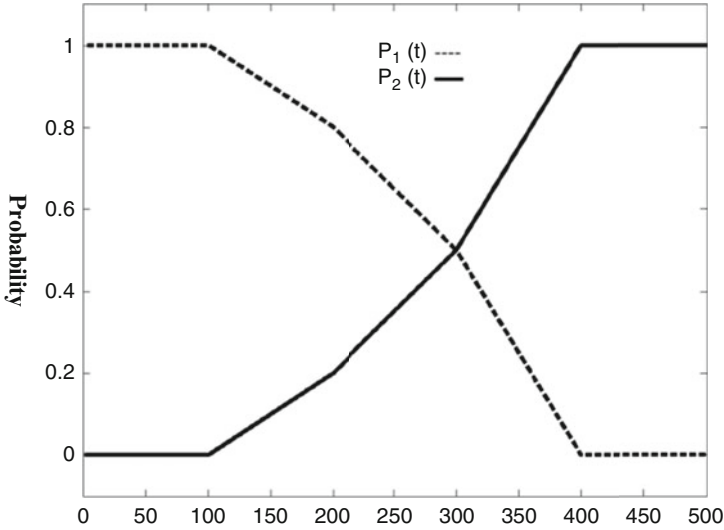


Fig. 2.14 Probability P_1 of occurrence of patterns from old concept and the probability P_2 of occurrence of patterns from new concept for the case of probabilistic gradual drift

occupied by the old concept in the feature space. Therefore, patterns belonging to both old and new concepts will cohabite.

Let us take the user preferences for document retrieval. He may change completely his search criteria for documents. Therefore, the new documents do not share any similarity with the old ones. In other words, no old document belongs to his preferences anymore. He may also change only some of his criteria for document search. In this case, some old documents remain belonging to his preferences.

2.4.3 Drift Influence Zones

Concept drift can be global (see Fig. 2.15b) or local (see Fig. 2.15c) according to the impacted zone of the feature space by the drift.

Local concept drift is defined as changes that occur in some regions of the feature space. Hence, the time required to detect the local drifts can be arbitrarily long. This is due to the rarity of data samples belonging to the new concept since both old and new concepts cohabite. Moreover due to this cohabitation between old and new concepts, data samples generated from the new concept can be considered as noises, which makes the model unstable. Hence, to overcome the instability, the model has to (1) effectively differentiate between local changes and noises and (2) deal with the scarcity of data samples that represent the local

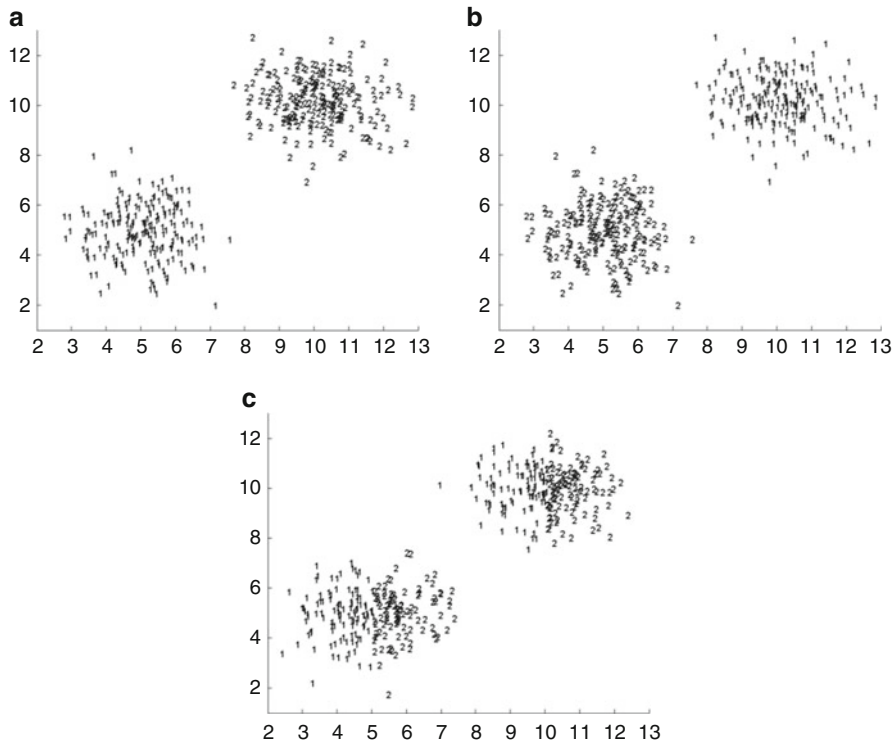


Fig. 2.15 Difference between drifts of partial and global severity. (a) Initial two classes W_1 and W_2 . (b) Global drift impacting all the regions of both classes. (c) Local drift impacting local zones of the regions occupied by both classes

drift in order to effectively update the learner. The global concept drift is easier to detect since it affects the overall feature space. In such case, the difference between the old and the new concept is more noticeable, and the drift can be earlier detected. This is due to the fact that the old concept will not cohabite anymore with the new concept.

Example 2.8: Local Concept Drift Let us take the example of Fig. 2.16. Figure 2.16.a shows the decision boundary of the initial classifier built using the data samples of the batch B_1 . In the second batch B_2 , a local drift occurred impacting a partial zone of the feature space between the two classes (see Fig. 2.16b). Therefore, only the patterns located in this zone will be impacted by the drift, while the other patterns keep their initial classes labels. The decision boundary will be updated by taking into account only the patterns impacted by this local drift (see Fig. 2.16b). In the batch B_3 , another local drift occurred. A new update of the decision boundary is required in order to maintain the classifier performance as we can see in Fig. 2.16c.

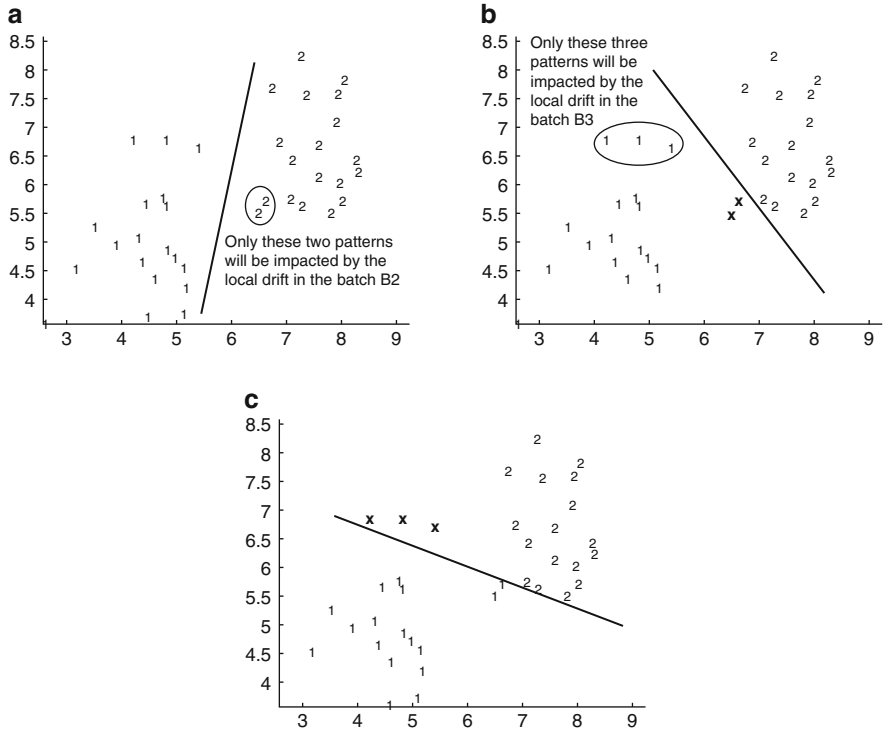


Fig. 2.16 Local drift impacting the feature space. The decision boundary of the initial classifier built using the data samples in the batch B_1 (a), after a first local drift in the batch B_2 (b) and a second local drift in the batch B_3 (c)

If we consider that class W_1 in Fig. 2.16b represents the viruses which are not resistant against antibodies while W_2 represents the viruses resistant to these antibodies, then some viruses of W_1 may develop a resistance against antibodies and become resistant while the other viruses remain as before not resistant. Therefore, this drift is just local drift concept.

2.4.4 Drift Occurrence Frequency and Recurrence

A concept may suffer from several drifts over time. If these drifts occur within regular time intervals, then they are called periodic drifts. Their occurrence frequency can be measured as the inverse of the number of time steps between two successive starts of drift. If these drifts occur in random or irregular time intervals, then they are called aperiodic drifts. If a concept suffers from the same drift at different time instances, then they are called recurrent drifts. In this case, concepts

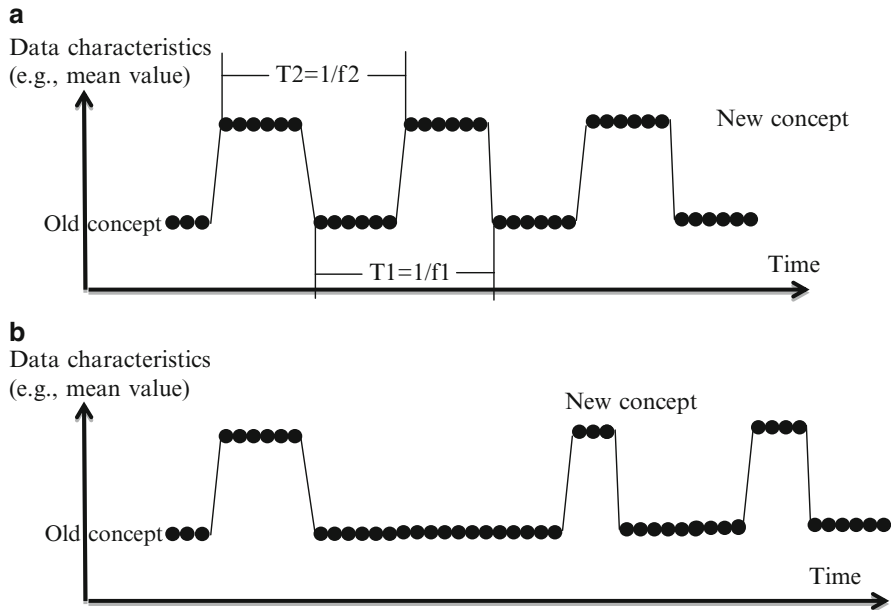


Fig. 2.17 Cyclic (periodic) recurrent drift (a) and acyclic (aperiodic) recurrent drift (b). f_1 is the occurrence frequency of the old concept, while f_2 is the occurrence frequency of the new concept

previously active may reappear after some time. Recurrent drifts may have *cyclic* or *acyclic* behavior:

- Cyclic recurrent drift (see Fig. 2.17a) may occur according to a certain periodicity or due to a seasonable trend. For instance, in electricity market, the prices may increase in winter due to the increase of demand and then return to previous level in the others seasons. The weather prevision is another example for cyclic recurrent drift where the prevision rules change in cyclic manner according to the active season.
- Acyclic recurrent drift (see Fig. 2.17b) occurs in aperiodic or random time intervals. For instance, the electricity prices may suddenly increase due to the increase of petrol prices (because of a political or economic crisis) and then return to previous level when petrol prices decrease.

It is worth underlining that when an old concept reappears, it may not be completely similar to its initial case. As an example, let us take a machine with two classes representing the normal and failure operation conditions. In the initial case, there is only the normal class. When the machine fails, a new concept occurs which is the failure class, and the old concept disappears which is the normal class. When the machine is repaired, it returns to the normal class (old concept). However, according to the efficiency of the maintenance actions and to the machine degradation status, the machine may not return completely to the class of normal operation conditions.

2.4.5 *Drift Predictability*

The predictability criterion was initially used in [18]. It indicates whether the drift is completely random or follows a pattern. Therefore, a drift is predictable if it follows a certain mechanism or set of rules. An example of predictable drifts is the weather forecasting. The change in the prediction rules is predictable according to the change in seasons.

A drift is unpredictable when its occurrence is random which does not follow any mechanism or rule. Example of unpredictable drifts is the occurrence of faults. Indeed, it is impossible to predict the occurrence of a fault before its occurrence. It can occur at any time and in different contexts.

It is interesting to consider the predictability of a drift for two reasons. First, when a drift is predictable, it is easier to understand its origins and expect its future effects (achieve a prognostic function). Second, a predictable drift can be accurately handled with a minimum delay of detection and false alarm rate, which is a desirable property in many real-world applications.

2.5 Drift Concept in Real-World Applications

There are multiple application domains in which concept drift plays an essential role. For these applications, the machine learning and data mining methods used to build a model (learner, classifier, etc.) for prediction or classification must take into account the concept drift in order to maintain the model performance and accuracy. However, in real-life applications, the concept drift may be complex or diverse in the sense that it presents time-varying characteristics. As an example, a concept drift can be recurrent, gradual, and local at some time instances, and then it becomes abrupt and global at other time instances. Hence in reality, often mixtures of many characteristics of drift can be observed during the transition phase of concept drift [21].

In [19, 21], the applications, where concept drift problem is relevant, are classified into four families:

- Monitoring against adversary actions as intrusion detection [22] or for management as traffic management and control [23]
- Personal assistance and information as recommender systems [24], customer profiling [25], and spam filtering [26]
- Predictions for decision making as evaluation of creditworthiness [27], electricity prices prediction [3], and sales prediction [28]
- Ubiquitous environment applications including a wide spectrum of moving stationary systems which interact with changing environments as moving robots [29] and smart house appliances [30]

In order to design an efficient self-adaptive model for an application where concept drift is related, the following points must be determined:

- Objective of the model: classification or regression
- Sources of the drift: environmental, system itself, or both
- Characteristics of the drift: speed, severity, predictability, etc.
- Speed of learning and required data loads
- Required accuracy and costs of mistakes
- Availability of true labels and kind of feedback
- Data samples balanced or imbalanced

As we will see in Chap. 3, these points allow to understand the drift phenomenon of the data streams generated by the application and therefore to guide the choice toward the suitable methods and tools to design an efficient self-adaptive learning scheme.

Example 2.9: Characterizing the Concept Drift Related to the Problem of Fault Diagnosis of a System Fault diagnosis aims at deciding at each instant whether a system works in normal operation conditions or a failure has occurred. The occurrence of a fault entails a drift in the system's normal operation conditions. This application has the following aspects:

- *Objective of the model* is the classification. The model is a classifier able to assign a new pattern, representing the current operation conditions, to one of two classes: normal or failure.
- *Sources of the drift* are exogenous caused by the system environments as a cut of physical connection between two switches because of an external action or the degradation of a service quality over time due to its wearing or the accumulated pollution, etc.
- *Characteristics of the drift*:
 - Two types of faults may occur: permanent and intermittent faults. Permanent fault can be either abrupt or gradual. If it is abrupt, then the drift is a shift in the system operation conditions from normal to faulty. While if it is gradual, then the drift is a degradation in the system performance. In this case, it is a continuous or incremental drift. If the fault is intermittent, then the drift is gradual probabilistic since patterns representing both normal and faulty classes cohabite. Then the number of fault patterns increases over time until the failure takes over completely.
 - The drift speed in both cases (gradual probabilistic and gradual continuous) depends on external factors generating these degradations, as the rate of pollution, moisture, temperature, etc.
 - The drift is acyclic recurrent since the fault may occur at any time and may be eliminated thanks to the maintenance actions.
 - The fault may be local (low severity) and/or global (high severity) impacting partially or completely the feature or instance space. As an example, a small degradation will generate a local drift since the system keeps an acceptable

performance which is not far from the nominal one. Example of local drift is the case where a pump is partially failed off or failed on. While an abrupt and severe fault generates a global drift as the case for a pump failed off or failed on completely.

- *Speed of learning* must be fast since the decision about the system status (normal/faulty) must be taken online in order to limit the fault adversary impacts.
- *Required accuracy and costs of mistakes* depend on the application criticism as well as the impact of faults on the system performance. For instance, if the system is a nuclear reactor, then the required accuracy of detecting faults is very high.
- *Availability of true labels (normal/fault)* is delayed. The true labels whether the system was in normal or failure operation conditions come available only after certain time (inspection, maintenance). These labels are hard; the system is either normal or faulty.
- *Data samples are highly imbalanced* since the data samples belonging to normal operation conditions are much more bigger than the ones representing a fault.

Learning from Data Streams in Dynamic Environments

Sayed-Mouchaweh, M.

2016, VIII, 75 p. 44 illus., 43 illus. in color., Softcover

ISBN: 978-3-319-25665-8