An online learning technique for coping with novelty detection and concept drift in data streams

Eduardo J. Spinosa¹, André Ponce de Leon F. de Carvalho¹, and João Gama²

¹ University of Sao Paulo (USP), Institute of Mathematical and Computer Sciences (ICMC), Caixa Postal 668, 13560-970, Sao Carlos, SP, Brazil
ejspin@icmc.usp.br, andre@icmc.usp.br

² University of Porto (UP), Artificial Intelligence and Computer Science Laboratory (LIACC), Rua Campo Alegre, 823, 4150, Porto, Portugal
jgama@liacc.up.pt
www.liacc.up.pt/~jgama

Abstract. To learn time-dependent concepts from streams of examples is one of the greatest challenges in machine learning. The ability to identify novel concepts as well as to deal with concept drifts are two important tasks in this scenario. This paper presents a technique based on the k-means clustering algorithm that treats these tasks as parts of a single learning strategy. Early experimental results are discussed and they inspire further investigation.

1 Introduction

Traditional inductive Machine Learning (ML) techniques attempt to extract knowledge from data by analyzing a set of examples. In supervised learning, each example is composed of a set of values, corresponding to attributes believed to be relevant to the problem being analyzed, and a label, indicating a class to which the example belongs. In the training phase, a model is built to represent the data, based on the examples of a denominated training set. Later, on a testing phase, the model is used to classify new unseen examples using the knowledge previously acquired in the training phase. The accuracy of the technique is measured on the test phase by comparing the predicted class to the correct class label of each example, that must be supplied for this purpose.

In a traditional approach to inductive ML, the model which represents the data is static, i.e., it remains the same after the training phase is completed. However, data distributions in real problems are hardly ever constant. As in life, concepts and definitions are continuously changing. Biologic brains possess an extraordinary capability for molding its structure to change and incorporate knowledge. For machines, however, this is a much more complicated task,
and it possibly represents the most important challenge currently faced by ML researchers.

The problem of incorporating the time variable into ML involves a number of issues that have been studied, sometimes individually, under different research fields. Works that explore novelty detection [5] or anomaly detection, for instance, try to identify new elements that might represent a novel concept yet to be learned. It is a field closely related to that of outlier detection, in statistics [1]. To be able to detect novelty, the technique must allow the learning of a single target or normal concept, which leads to the term one-class classification [6] or single-class classification. The absence of negative examples, which would represent the unknown novelty in this scenario, makes it hard for the induction algorithm to establish adequate decision boundaries and thus avoid the extremes of underfitting and overfitting. This problem has also been studied under the term learning from positive-only examples. Since the main objective is to learn a representation of how the data is distributed, some works name it a data domain description.

Besides recognizing novelty, to be capable of dealing with the time variable, a ML technique must also be able to identify changes occurred in the known or normal concept. This problem has been studied under the term concept drift [2].

In this ever-changing scenario, the definitions of normal, known, positive or target, as well as those of anomaly, novelty or outlier, are intrinsically time-dependent. These concepts are subject to an equally time-dependent model, that must be induced in a certain manner so as to reflect the data distribution at any time, as accurately as possible. Techniques that are capable of updating the model as new data is received coined the term online learning, which is clearly a precondition for the development of ML techniques capable of dealing with time-dependent problems.

Additionally, the massive amount of data involved in most real problems when the time variable is considered, motivates further considerations. The idea of static training and testing data sets no longer applies, once examples are continuously received, treated and discarded in a continuous flow expressed by the term data stream. Depending on the rate at which data is received, performance constraints also apply, which leads to the construction of simpler algorithms capable of rapidly classifying new examples and updating the model as the learning continues.

This paper presents the current state of development of an online learning technique aimed mainly at identifying and dealing with both novelty and concept drift. The next section describes its strategy and briefly discusses related works. Section 3 presents and analyzes the results of early experiments. Section 4 summarizes the conclusions which can be drawn at this stage and provides insight into future investigation.
2 Proposed technique and related works

The technique proposed in this paper considers novelty detection from a one-class classification perspective. This means that data distribution is represented by a model that is learned based on positive-only examples, belonging to a concept we call normal. At each time, examples that are incompatible with the current normal concept are named unknown.

2.1 Distinguishing between normal and unknown

The normal model, which represents the normal concept, is built based on the k-means clustering algorithm. Given a set of examples belonging to the normal concept and the number of clusters \( k \), the k-means algorithm produces \( k \) sets of examples denominated clusters and returns the position of each cluster’s centroid. Then, for each cluster, we calculate the maximum distance between examples and the centroid, i.e., the distance between the centroid and the example from which it is farthest. This allows us to establish a decision boundary for each cluster. The union of the boundaries of all clusters is the global decision boundary that defines the normal concept. A new unseen example that falls inside this global boundary is considered normal; otherwise, it is labeled unknown.

In the literature, this phase is frequently referred to as novelty detection. However, we use the term unknown instead of novelty, since it is our understanding that a single example should not be considered novelty, in the sense of a novel concept. It could simply be noise. Due to lack of evidence, we let unknown examples wait in a separate data set denominated short term memory of unknown data, which is frequently monitored for the formation of new clusters that might indicate two possible conditions that are treated by this strategy: novelty and concept drift.

2.2 Identifying novelty and concept drift

To monitor the formation of clusters in the short term memory of unknown data, we use the k-means clustering algorithm for generating \( k \) candidate clusters. For a candidate cluster to be considered valid, which might indicate a concept in our approach, it must fulfill the following requirement. First, we calculate the overall arithmetic mean of the mean distances between centroids and respective examples of all clusters of the normal model. Then, we compare this value with the mean distance between the centroid and examples of the candidate cluster. If the value for the candidate cluster is lesser than or equal to the one obtained for the normal model, the candidate cluster is consider valid. This restriction aims at selecting clusters whose density is not lower than that of the normal model.

Once a candidate cluster has been validated, our strategy considers two possible explanations for the emergence of that cluster: it either represents a novel concept, denominated novelty, or it is an evidence that the normal concept is undergoing a change, which is referred to as concept drift. For that decision, we consider that clusters which are closer to the boundaries of the normal model
are more likely to appear due to a drift occurred in the normal concept. On the other hand, validated clusters appearing far from the normal concept might be due to the emergence of a novel concept, or novelty.

To establish this limit, our strategy calculates $d_{\text{max}}$, the maximum distance between the centroids of the normal model and a global centroid, which is the centroid of a set that contains all the centroids of the normal model. The distance $d_{\text{max}}$ indicates how far the farthest centroid is from the global centroid. We consider $d_{\text{max}}$ as a threshold for distinguishing between concept drift and novelty. If the distance between the global centroid and the centroid of the new validated cluster is lesser than or equal to $d_{\text{max}}$, the emergence of that new cluster is attributed to concept drift; otherwise, a novel concept or novelty is identified.

### 2.3 Updating the normal model

The proposed technique periodically updates the normal model to ensure that it represents the current definition of the normal concept. In order to do that, it maintains a data set denominated short term memory of normal data, which contains (1) examples that have been identified as normal by the algorithm described in Subsection 2.1, and (2) examples of validated clusters that have been identified as concept drift, according to the algorithm described in Subsection 2.2. The update is performed whenever a certain number or examples, defined by the user, are waiting in the short term memory of normal data. By incorporating new examples to the normal model, the technique is capable of learning in an online manner, which is a precondition for it to work with data streams.

### 2.4 Reducing the influence of parameters

In any ML technique, parameters that control the learning process are capable of, among other aspects, establishing the degree of generalization of the model. The more a technique depends on parameters, the less robust it will be to the undesirable extremes of underfitting and overfitting. Given that perspective, we attempt to minimize the number of parameters and their influence on the decision process by using thresholds calculated based on distances between examples and centroids that depend solely on the data, as described in Subsections 2.1 and 2.2.

The number of clusters $k$ is required by the k-means clustering algorithm. However, experiments have shown that simply using a fixed $k_{\text{nor}}$, for building and updating the normal model, and a fixed $k_{\text{newcl}}$, for monitoring the formation of new clusters, becomes inadequate after a certain time. As data sets grow with the addition of new examples, the constant numbers of clusters $k_{\text{nor}}$ and $k_{\text{newcl}}$ become relatively smaller. With time, clusters tend to grow excessively, reducing the level of detail of the model and, thus, leading to overgeneralization. To eliminate that effect, we created a single higher level parameter to replace both $k_{\text{nor}}$ and $k_{\text{newcl}}$: the number of examples per cluster $n_{\text{excl}}$. Every time the k-means algorithm is required, a different value of $k$ is calculated in order to maintain the number of examples per cluster $n_{\text{excl}}$. The greater the number of
examples, the greater the number of clusters will be. This strategy leads to an online adaptation of $k$, which ensures that the level of detail is maintained both in the normal model and for monitoring the formation of new clusters among examples identified as unknown.

A few secondary parameters allow a customization of the technique. The user can specify the number of examples used to generate the initial model $n_{ini}$. To reduce computational cost, it is also possible to determine that the updating of the normal model should only take place whenever the short term memory of normal data contains a certain number of examples $n_{wait}$. And for applications in which the algorithm remains online for a great period of time, it is possible to forget older examples and use only the latest $n_{limit}$ to update the normal model.

2.5 Related works

To the best of our knowledge, a strategy for the treatment of both novelty detection and concept drift has not yet been proposed in the literature. Differently from most of the works under novelty detection and similar terms, which consider novelty as examples that do not fit a certain data distribution, we regard novelty as a concept. From our perspective, the term novelty is represented by a cluster of examples that comply with certain restrictions and, thus, might indicate the appearance of a new concept.

The k-means clustering algorithm has been previously used to construct a novelty detector [6], with an approach which is, however, very different from the technique proposed in this paper. In that work, the decision boundaries are positioned based solely on a fixed parameter, called rejection rate, that forces a certain percentage of the normal examples to stay outside the borders.

3 Experiments

The technique proposed in this paper has been implemented in R [4]. This section reports and analyzes early results obtained with the Iris data set, from the UCI Machine Learning Repository [3]. It consists of 150 examples, evenly distributed among 3 classes: Iris-setosa, Iris-versicolor and Iris-virginica. Class Iris-setosa is linearly separable, and the other two are harder to distinguish. Since there are only 4 attributes, data can be clearly visualized in 2 dimensions, which facilitates the understanding of the learning process.

The nature of this work, i.e., one-class classification, requires that the learning be performed based on examples of a single class. Therefore, each of the three classes of the Iris dataset has been considered the normal concept at a time.

The proposed technique is nondeterministic, since (1) the k-means algorithm produces different clusters each time, and (2) examples are shuffled prior to execution. For that reason, each of the experiments has been executed 10 times in order to obtain average results.

In the following experiments, 30 examples ($n_{ini}$) were used to generate the initial normal model, and the number of examples per cluster $n_{excl}$ was set to 10.
3.1 Evaluating the distinction between normal and unknown

After the initial normal model is induced, new unseen examples are presented. The algorithm described in Subsection 2.1 classifies each of them as either normal or unknown. A normal example classified as unknown is denominated false-unknown. The opposite situation, called false-normal, happens when an unknown example is classified as normal. When a certain class is considered normal, the two remaining classes are labeled unknown. In the literature, this phase is frequently referred to as novelty detection.

Table 1 presents mean error rates due to false-unknown and false-normal obtained at each of the experiments. The values represent the arithmetic mean and standard deviation of 10 executions.

<table>
<thead>
<tr>
<th>Class considered the normal concept</th>
<th>Error rates due to false-unknown</th>
<th>Error rates due to false-normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris-setosa</td>
<td>0.08 0.06</td>
<td>0.00 0.00</td>
</tr>
<tr>
<td>Iris-versicolor</td>
<td>0.16 0.11</td>
<td>0.13 0.05</td>
</tr>
<tr>
<td>Iris-virginica</td>
<td>0.13 0.07</td>
<td>0.32 0.10</td>
</tr>
</tbody>
</table>

Table 1. Error rates due to false-unknown and false-normal.

3.2 Evaluating the identification of novelty and concept drift

Examples classified as unknown remain in the short term memory of unknown data and are continuously monitored for the formation of valid clusters, according to the algorithm described in Subsection 2.2. The unpredictability inherent to the k-means clustering algorithm makes it hard to develop an evaluation strategy. The fact that a certain novelty or concept drift has not been identified at a certain time $t_i$ does not mean that there is the same chance that it will not be found at time $t_{i+1}$.

To visualize the appearance of new clusters directly, the user has the option to enable a series of comprehensive bidimensional plots that represent snapshots of the online learning process. Figures 1 and 2 display two examples of these sets of plots, in which the third and the forth attributes of the Iris data set were selected as coordinates. Each figure is composed of six plots, described as follows.

(a) **Model.** Clusters whose union represent the normal model. Examples are marked by circles and centroids by bolded circles. A different color is used for each cluster.

(b) **Normal.** New unseen normal examples. The ones that have been correctly classified as normal are marked by circles, and those that have been wrongly classified as unknown (false-unknown) are marked by filled circles.
(c) **Unknown.** New unseen *unknown* examples. The ones that have been correctly classified as *unknown* are marked by squares, and those that have been wrongly classified as *normal (false-normal)* are marked by filled squares.

(d) **Candidate clusters.** Candidate clusters being considered at a certain time. These clusters have not yet been subjected to the restrictions described in Subsection 2.2.

(e) **Concept drift.** Validated clusters identified as *concept drift.*

(f) **Novelty.** Validated clusters identified as *novelty.*

---

**Fig. 1.** Example of a snapshot of the online learning process, taken when a new validated cluster is identified as *novelty.* In this experiment, class *Iris-versicolor* was considered the *normal concept.*

Figure 1 displays a snapshot taken when a new validated cluster is identified as *novelty.* In this experiment, class *Iris-versicolor* was considered the *normal concept,* as seen in Plot (a). Plot (c) shows that class *Iris-setosa,* on the lower left part, and class *Iris-virginica,* on the upper right part, are correctly classified as *unknown* with only a few mistakes due to the fact that classes *Iris-versicolor* and *Iris-virginica* are harder to distinguish. Plot (d) shows candidate clusters generated by examples classified as *unknown.* One of these clusters is validated and identified as *novelty,* shown in Plot (f).

Figure 2 shows a new validated cluster being identified as *concept drift.* In this experiment, class *Iris-virginica* was considered the *normal concept.*

The Iris data set does not actually present cases of *concept drift.* That explains why our technique very rarely identified this occurrence. In the example
presented in Figure 2, concept drift was only detected because, since examples of class Iris-virginica, which was considered the normal concept, are partially mixed with those of class Iris-versicolor, the algorithm inferred that Iris-virginica, the normal concept, was drifting to Iris-versicolor. In Plot (a), it is possible to verify, colored in yellow, that examples belonging to this new cluster, identified as concept drift and shown in Plot (e), have been used to update the normal model.

Although useful, such a bidimensional visualization is hardly ever applicable when dealing with real data sets with a great number of attributes and examples. In order to evaluate the ability of the proposed technique to identify novelty and concept drift, a higher level plot that shows how classes are distributed among the three data sets maintained by the algorithm is also generated. Figures 3 and 4 show examples of typical distributions obtained by the end of the learning process.

In Figure 3, when class Iris-setosa was considered the normal concept, the technique correctly identifies Iris-versicolor as novelty. In some executions of this experiment, class Iris-virginica has also been identified as novelty. Similar results were obtained when class Iris-versicolor was considered the normal concept.

In Figure 4, it is possible to observe that, even though the normal concept is represented by class Iris-virginica, a large number of examples of class Iris-versicolor have also been included in the normal data set, about 40%. It
Fig. 3. Example of class distribution among the data sets by the end of the learning process when class Iris-setosa was considered the normal concept. means that when the normal model is constructed based on class Iris-virginica, the nearby class Iris-versicolor is sometimes considered normal. This result is confirmed by a larger number of false-normal occurrences in this experiment, expressed by the 32 % error rate in Table 1.

Fig. 4. Example of class distribution among the data sets by the end of the learning process when class Iris-virginica was considered the normal concept.

4 Conclusion

This paper proposes a learning technique, based on the k-means clustering algorithm, that is capable of coping with both novelty detection and concept drift.
The strategy is able to differentiate examples of a known *normal concept* from others, here denominated *unknown*. In the literature, this is frequently referred to as *novelty detection*. However, in this work we propose a different definition of *novelty*, which denotes a *novel concept* represented by an emerging cluster of examples that has been validated by the technique.

In addition to the ability of detecting *novel concepts*, the strategy can also identify changes occurred in the *normal concept*, which is denoted by the term *concept drift*.

By updating the *normal model*, the proposed technique is able to work in an online fashion, a precondition to its future application to real problems in which examples are continuously received, treated and discarded, in a flow known as *data stream*.

Results obtained so far inspire further investigation, including experiments with a greater number of larger data sets (artificial and real) that present novel and changing concepts. Comparison with other online learning strategies is also regarded as a relevant step to the consolidation and development of the proposed technique.

To empower machines with the abilities to change and to incorporate knowledge is one of the greatest challenges currently faced by ML researchers. We believe that a way to confront such defiance is by approaching the detection of both *novelty* and *concept drift* by means of a single strategy.

### 4.1 Acknowledgment

The authors wish to thank the financial support of the National Council for Scientific and Technological Development (CNPq), a foundation linked to the Ministry of Science and Technology (MCT) of Brazil.

### References