Environmental Data Stream Mining through a Case-Based Stochastic Learning Approach

Fernando Orduña Cabrera and Miquel Sànchez-Marrè[1]

forduna@gmail.com,miquel@cs.upc.edu

[1]Intelligent Data Science and Artificial Intelligent Research Centre (IDEAI-UPC)
Knowledge Engineering & Machine Learning Group (KEMLG), Dept. of Computer Science,
Universitat Politècnica de Catalunya (UPC), Barcelona, Catalonia

Abstract. Environmental data stream mining is an open challenge for Data Science. Common methods used are static because they analyze a static set of data, and provide static data-driven models. Environmental systems are dynamic and generate a continuous data stream. Dynamic methods coping with the temporal nature of data must be provided in Data Science. Our proposal is to model each environmental information unit, timely generated, as a new case/experience in a Case-Based Reasoning (CBR) system. This contribution aims to incrementally build and manage a Dynamic Adaptive Case Library (DACL). In this paper, a stochastic method for the learning of new cases and management of prototypes to create and manage the DACL in an incremental way is introduced. This stochastic method works with two main moments. An evaluation of the method has been carried using a data stream of air quality of the city of Obregon, Sonora. México, with good results. In addition, other datasets have been mined to ensure the generality of the approach.

Keywords: Data Science, Data Stream Mining, Dynamic Case Learning, Stochastic Learning, Case-Based Reasoning, Air Quality Detection, Environmental Modelling.

1 Introduction

Data science discipline has a main goal to obtain effective knowledge from data. This effective data analysis must be translated into useful knowledge and information for decision support. In the usual application of data mining techniques, there is a gap between the mined models and the effective knowledge needed for a reliable decision-making process. Many works in the literature have applied some data mining methods to build assessment models and/or predictive models in environmental domains. The main problem on most of the approaches is that assessment and predictive models are based on static supervised environmental databases gathered from a concrete period of time. Therefore, on one hand the models are static and cannot capture the dynamic nature of such environmental data streams, and on the other hand, the number of different environmental conditions or states are determined *a priori*. These models are not very reliable for the stakeholders' decision-making process. The temporal component of data must be taken into account jointly with quantitative and qualitative data, and the number of different environmental conditions or states must be discovered through the mining of the environmental data stream. The approach presented in this paper integrates these different information sources and build an incremental data-driven method based on a case-based stochastic learning approach for assessment of environmental conditions. To illustrate the approach, it has been applied to a case study on air quality assessment conditions.

In recent years, the problem of mining data streams has grown the attention of many researchers. Many real-world applications generate data continuously. For example, in network monitoring, environmental data sensors, telephone record calls, multimedia data, customer transactions, customer click streams, and so on. Advances in technology have facilitated new ways of continuously collecting data. In many applications, the volume of such data is so large that it may be impossible to store the data on disk. Furthermore, even when the data can be stored, the volume of the incoming data may be so large that it may be impossible to process any particular record more than once. Therefore, many data mining and database operations such as classification, regression, clustering, frequent pattern mining and indexing become significantly more challenging in this context (Aggarwal, 2007). The monitoring of many events in real time produces much information. In recent years, data stream mining field has grown rapidly. In (Hulten & Domingos, 2001) outlined some desirable properties for learning tasks in data streams: incrementality, constant time to process each example, single scan over the training set, and taking *concept drift* into account. Learning from data streams require *incremental learning* algorithms that take into account the problem of *concept drift*: the underlying concept or distribution of the data can change over time, and the mined models should be aware of the changes, and adapt themselves to the changes.

Our proposal is to model the new environmental information timely generated as a new case/experience in a Case-Based Reasoning (CBR) system. CBR (Richter & Weber, 2013, López de Mántaras *et al*., 2005) solves new problems/cases using old similar solved problems/cases in the same domain. The designed method aims to incrementally learn the new cases, build new prototypes of cases, and store the cases in the most similar prototype. A prototype of cases is a generalization of similar cases, which represents a similar environmental situation (i.e., an air quality environmental condition). This contribution aims to incrementally create and manage a Dynamic Adaptive Case Library (DACL) as a way to strengthen its structure. The proposed method implements a DACL, which can be used as a predictive system for new environmental data situations, warning the experts about dangerous situations for the citizen. In this paper, a stochastic method for the learning of new cases and management of prototypes to create and manage the DACL in an incremental way is introduced. This stochastic method works with two main moments. The first moment guides the learning of new cases and decides where to store the cases. The second moment evaluates the candidate prototype, and select it or builds a new prototype. This way, through this data-driven process, an automatic, adaptive and dynamic system for supporting decisions can be deployed.

1.1 Background knowledge

In machine learning literature, several works have addressed the problem of learning from data streams (Gama 2010; Gama & Gaber, 2007), and other works studied the time changing concept problem (Hulten et al., 2001; Klinkerberg & Joachims, 2000; Maloof & Michalsky, 2000; Kubat & Widmer, 1995). Most common techniques used are temporal windows, which determines the training set for the learning algorithm, and the weighting of examples, which attempts to decrease the relevance of the older examples, and increase the relevance of the new ones. Also there are some mixture techniques. Some authors (Klinkerberg, 2004; Klinkerberg & Renz, 1998; Widmer & Kubat, 1996) propose the use of adaptive time windows in order to minimize the generalization error of the classification models.

Continuous problem domains (i.e., domains where cases are generated from a continuous data stream) require different underlying representations and place additional constraints on the problem solving process. (Ram & Santamaria, 1997) define three characteristics where the problem domain is continuous, and those are: First, they require *continuous representations*. Second, they require *continuous performance*. Third, these problem domains require *continuous adaptation* and learning. Reasoning about continuous domains is not an easy task. Moreover, this is a domain where CBR can rapidly extend its benefits because data is systematically collected for its analysis. A CBR system that continuously interacts with an environment must be able to create autonomously new situation cases (new concepts or clusters) based on its perception of the local environment in order to select the appropriate steps to achieve the current mission goal (Haris & Slobodan, 2005), but a general framework is still missing. Some systems that use case-based methods in continuous environment are described in (Urdiales *et al*., 2006; Kruusmaa, 2003; Ram *et al*., 1997).

There are two other central problems derived from the continuous nature of some domains. First of all, the *size of the case library* could grow very fast as the CBR system is learning new cases without an extensive improvement in the competence of the system, as pointed out in (Miyashita & Sycara, 1995). Two natural human cognitive tasks appear as the solution to these problems: forgetting (Keane & Smith, 1995) and sustained relevant learning (Sànchez-Marrè *et al*., 1999). On the other hand, learning many cases could provoke an *overhead in the case library organization*. As new cases are stored in the case library, it will be necessary to update the case library organization (Meléndez *et al*., 2001).

Sànchez-Marrè in (Sànchez-Marrè et al., 2000) introduced the idea of using prototypes for improving the retrieval of cases in a static multiple case library. This multiple case library was composed of expert-defined prototypes and its corresponding hierarchical case libraries.

Finestrali and Muñoz in (Finestrali and Muñoz-Avila, 2013) implements a stochastic explanation to determine the learning goal while plays Wargus. This game is an example of a stochastic domain. Their studied the problem of explaining events in stochastic environments using Case-Based Reasoning (CBR). The center of their approach has three ideas: (1) Using the notion of Stochastic Explanations (2) Retaining as cases (event, stochastic explanation) pairs when such unexpected events occur. (3) Learning the probability distribution in the stochastic explanation as the cases were retrieved. Their proposal is novel and somewhat similar to our work. In our work, the stochastic method is the core of the learning of new cases and building new prototypes, while they use Q-Learning to give the stochastic explanation of the learning and labelling of the case.

Stochastic clustering methods have been introduced in clustering field like in the case of Chuan in (Tan et al., 2010), where their proposal examines a practical stochastic clustering method that has the ability to find clusters in datasets without requiring users to specify the centroids or the number of clusters. Their experimental setup confirms that the proposed method performs competitively against the traditional clustering methods in terms of clustering accuracy and efficiency. In their work, an estimation of the similarity thresholds for *n* items is computed. The estimation of similarity thresholds is widely used. For instance see (Orduña Cabrera, 2016).

In (Tan *et al*., 2010), authors implement a stochastic method where the classification process and building of the clusters are the aim. This is done without human interaction. They use the time variable to evaluate the probability of belonging to some cluster. In our proposed method, we use a time variable to indicate the time of the acquired data. Both methods learn cases and store it, but our approach differs from theirs in the method of learning the cases. In our case, the proposal is focused on the representative prototypes. In addition, the learning of cases in our proposal is conditioned to accomplish with the maximal acceptable dispersion.

Air quality assessment is important for air pollution control and environmental management. Air quality is an important concern over the world, and especially in urban areas. Local city authorities and governments are responsible for the continuous monitoring and improvement of the urban air quality. There are several contributions in the study of the air pollution. These contributions study different aspects related to the air contamination. Some of this research works are (J.I. Halonen et al, 2016; Costabile and Allegrini, 2007). In (J.I. Halonen et al,. 2016) the effect of long-term exposure to traffic pollution is studied. They study the associations between traffic pollution and emergency hospital admissions for cardio-respiratory diseases. In (Costabile & Allegrini, 2007) the study aims to analyze the relationship between air quality and air pollution from transport, and they develop a framework to understand its relationship. The effect of the air pollution and the noise of the traffic, are one of the main interest in the study of diseases produced for its high contamination levels. Some studies that evaluate the noise levels are (Gulliver *et al.*, 2015; De Coensel et al,. 2012; Tang and Wang, 2007). All these proposals analyze data acquired from special sensors and use scientific methods to translate from data to knowledge. Others studies are focused in building models (Vedrenne et al., 2014; Fallah-Shorshani et al,. 2015; Reis et al,. 2015) and computational applications (Wyat Apple et al,. 2011; Carslaw and Ropkins, 2012, Hill & Minsker, 2010) that help to understand the information acquired. Others proposals predict possible scenarios such in (Liu *et al*,. 2015). A different study including patterns of the air direction is detailed in (Thaker & Gokhale, 2016), where they investigate the effects of different urban traffic flow patterns on pollutant dispersion in different directions of the winds. However, all of them analyze static data and use static models for decision support, which do not reflect the dynamic nature of environmental data.

2 Methodology

2.1 Dynamic CBR

Case-Based Reasoning (CBR) systems solve new problems by retrieving and adapting the solutions to previously solved problems that have been stored in a case library. CBR is a very flexible reasoning paradigm, which can be used both as a classification/discriminant technique and as a regression/predictive technique. The most common approach to predict a class label is the use of a simple CBR scheme (k-nearest neighbor classifier), but it can also be used to predict numerical variables, in a regression problem. This flexibility makes it a powerful tool for data mining. Furthermore, CBR integrates a learning step in its basic reasoning cycle. This learning activity makes CBR to be very suitable to be used for dynamic learning purposes. CBR systems become more competent over time, because they learn from experience. CBR approaches can process a data stream with a fine-grained time window of length one. They can process tall the examples/cases one by one and adapt their model at each example (*on-line learning*).

The case learning task is one of the important steps in Case-Based Reasoning (CBR) systems, and especially, in domains where a large amount of information needs to be managed efficiently (data stream). The evaluation of environmental domains requires of acquiring and processing a large amount of data. These environmental data streams can be managed with dynamic strategies to support in making good decisions, as the outcome of the data science process.

A very important aspect related to unsupervised continuous domains is the *incrementality problem*. General CBR systems assume that the set of cases available for building the case library is fixed and available at the beginning (batch learning). Then, they build the memory indexing structures, like for instance, a *k*-d tree, decision/discriminant tree, etc. However, when a CBR system is facing an unsupervised continuous domain (data stream), the system should build and update the case library structure/s in an *incremental* way (incremental learning).

A Dynamic adaptive framework is proposed to improve the CBR system performance coping especially with reducing the retrieval time, increasing the CBR system competence, and maintaining and adapting the case library to be efficient in size, especially in continuous domains (data streams) [6]. The framework proposed works for reasoning and learning both in supervised domains and unsupervised domains.

At the beginning, the DACL structure is *empty,* and there is not an initial predefined number of clusters. There are neither initial prototypes nor clusters. When the first case processed of the data stream arrives, as there are yet no clusters, it is created the first cluster with just the first case, by now (we have now one cluster). The prototype of the first cluster is the same new case. In addition, this case is added to the discrimination structure associated to the new cluster, which now has just one case. As the new cases of the data stream are arriving to the system, the DACL applies the learning strategy. If the new cases are similar enough to some existing prototype, taking into account the two moments of the learning scheme: dissimilarity to prototypes (equation (1)) and dispersion of most similar prototype (equation (2)), they are assigned to that cluster. Then, the prototype of the cluster is re-computed, and the new case is stored in the corresponding discrimination tree associated to that cluster. On the other hand, if new cases are not similar enough to any prototype (it does not fall within the dispersion area of the most similar prototype), a new cluster is created with that new case as the unique case of a new cluster, and it is the prototype of that new cluster. The new case is stored in a new empty *k*-d tree bind to the new cluster.

This way, the number of clusters/prototypes is increasing until the entire data stream is processed. This way the number of clusters/prototypes is *automatically determined* in a dynamical, incremental and stochastic way, because the case learning randomly depends on the underlying probability distribution of the data stream and on the parameter gamma (dispersion relaxation), a different number of clusters/prototypes are discovered after processing the entire data stream.

Hence, the DACL approach provides both an incremental clustering technique, which discovers automatically the number of clusters, and an incremental predictive technique based on the set of sub-libraries (the indexing tree structures) associated to each cluster/prototype. To make a prediction for a new case, it just have to select the most similar prototype to the new case, and search for similar cases in the sub-library (*k*-d tree) associated to that prototype/cluster. The idea is that the most similar cases are the ones belonging to the same cluster than the new case.

Here in this paper, we have focused on the identification problem of environmental situations, i.e. in the problem of discovering the clusters in an incremental way, but also the predictive task has been assessed as explained later.

Figure 1 shows the library, after several cases processed, which is dynamically constructed with several sub-libraries. Each sub-library is organized hierarchically at the following levels:

- The Meta-Case: The meta-case ***Mc***, which is the prototype of a concrete cluster of cases.
- The Cluster: the set of cases represented by the prototype. In fact, this level is an abstract one, because the cases are stored at the next level of the indexing structures
- The Indexing Hierarchical structures: Represents the way that all the cases are organized in the sub-library (for instance, discriminant trees or *k*-d trees).

Figure 1 depicts in top level the Meta-Case. In this level could exist several number of Meta-Cases. Each one for each class type of data (i.e, the different subconcepts being discovered in the data science process). The lower level is the indexing strategy place; here the cases are stored but are organized using the proposed indexing techniques of (Orduña Cabrera and Sànchez-Marrè, 2013). This indexing technique is a NIAR *k*-d tree with a partial matching exploration technique, which showed the best performance both in time and accuracy (Orduña Cabrera, 2016). The retrieval process in a NIAR *k*-d tree is as follows: given a new case, the distance between the new case and all the Meta-cases must be computed. The most similar Meta-case will be selected, and its corresponding NIAR *k*-d tree, will be traversed for searching the most similar cases. The traverse follows the branch satisfying the condition on the attribute value associated to each node (≤, >) until a leave or leaves are found. NIAR *k*-d trees are also binary trees like standard *k*-d trees.

The building of the NIAR *k*-d trees is incremental. That means that each time a new case must be stored in a *k*-d tree, the tree has to be slightly updated in an incremental manner. NIAR k-d trees (Orduña Cabrera and Sànchez-Marrè, 2013) are built quite similar to the standard *k*-d tree, by selecting the discrimination attributes cycling through the list of attributes, but differs of standard *k*-d tree approach in the technique of selecting the split value for each attribute at the internal nodes. The proposed partition value (the Root) is the Nearest attribute value of an Instance to the Average value of the attribute values from the instances in the corresponding node (NIAR).

The DACL proposal in (Orduña Cabrera and Sànchez-Marrè, 2009) is illustrated in figure 1. A DACL is composed of a set of dynamically built case libraries to cope with the heterogeneity and complexity of real domains. It learns cases and organizes them into the dynamic cluster structures. The library is able to adapt itself to a dynamic environment, where new clusters, meta-cases, and associated indexing structures can be formed, updated, or even removed, according to the data stream distribution. DACL offers a possible solution to the management of the large amount of data generated in a continuous domain (data stream). The concern here is the introduction of a proposal of a stochastic learning method that helps building the core of a DACL that is called *the prototype or the meta-case*. The technique used here has a similar intention of using a stochastic method.

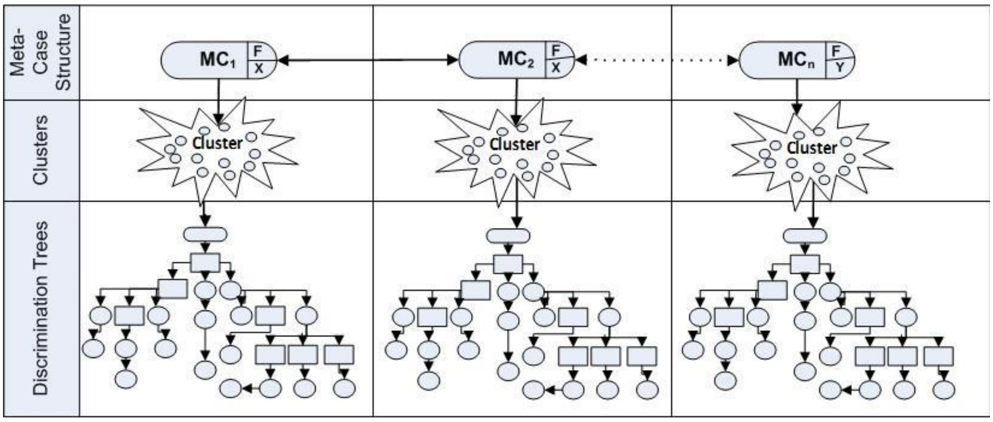


Fig. 1. DACL Structure after several cases processed, and having built *n* clusters and associated structures

2.2 Stochastic Learning Method

In stochastic processes, the first and second statistical moment of a signal or a data collection can detect whether the data is taken from a stationary process or not. With this information, and evaluating the density probability function, it could be known from which sample of the whole data collection (data stream) the data processed comes from.

The process could be stationary in certain periods and not stationary in other ones. The statistical moments (mean and variance/standard deviation) varies from sub-concept to sub-concept. This is the hypothesis used in this work and helps to decide to which cluster belongs a new data item according to the two statistical moments. If the new data processed is different from previous established clusters (characterized by the mean and variance or standard deviation), then a new cluster (a new state of the stochastic process) is dynamically created.

The processing of the data stream and the dynamic learning of the set of clusters constitutes a stochastic process. The probability of belonging to the different clusters is changing during the entire processing of the data stream. Depending on distribution probability of the cases in the data stream, the formation of the clusters is different (in number and content), changing the probabilities of belonging to the different clusters for a new case. Thus, the learning process is said to be stochastic.

The proposed method has the ability to learn Meta-Cases (*Mc*) as is depicted in figure 1. In this work, a *Mc* is described as following:

The Mc's are designed as the top level of the DACL strategy. This has two aims. The first is when a New Case (*Nc*) is being considered by the DACL. It has to learn it and store it in the best optimal way or decide not to store it; and second, when the best case has to be retrieved. This strategy should improve time and quality of the process avoiding an exploration in a wrong sub-library. *A Mc is a prototype of the entire cases belonging to the sub-library*. The *Mc* in DACL helps to find the most optimal sub-library where the cases have to be learnt. The *Mc* is an generalization of the cases stored in the sub-library. The *Mc* is built following the next *Mc building process*:

1) A case C^k is defined as $C^k =< att_1^k \dots att_n^k >$ where att_i ,i=1,..,n are the attributes describing the case C^k and where k is the current case of the total of m cases belonging to the corresponding Mc(k=1, ...,m).

2) A Mc^l is defined as $Mc^l =< att_1^l \dots att_n^l >$ where att_i, i=1,...,n is the new computed attribute as the average prototype value for continuous/numerical attributes or the most frequent value (mode) of discrete categorical values, according to the following formulas:

If att_i is numerical: $att_i^l = \frac{1}{m}\sum_{k=1}^{m} att_i^k$ and #cases(Mc^l)= m

If att_i is categorical: $att_i^l = mode(att_i^k)$ k = 1,...,m

A case is defined by its attributes: $< att_1, \dots, att_n >$. The strategy adds a new attribute att_{ts}, and then, a case is described as: $< att_{ts}, att_1, \dots, att_n >$. This new attribute is a *time stamp* ordering identifier (*ts*). The new attribute is initialized at $ts = 1$ and increases one by one. The increase occurs when a new case is stored in the sub-library.

This att_{ts} value is considered like an attribute in the algorithm. The attribute is used in both first and second moment of the stochastic method proposed. It is named as τ. τ plays the role of time, an ordered attribute, like in a normal stochastic method. The value of τ helps to increase the difference of cases, adding between them an increase value taken into account when first moment is computed (see equation (1)).

The method can be summarized as follows: when a new case arrives (*Nc*), the algorithm finds the most similar *Mc* to the *Nc* (most similar stochastic state based on the mean values). The most similar *Mc* to the *Nc* is the *Mc* at a minimum distance of *Nc* according to what is described in equation (1). Afterwards, if the dispersion (deviation) of the *Nc* regarding the mean values of the most similar *Mc* (ρ) is less than the relaxed (by factor γ) deviation of the cases represented by ρ (as described in equation (2)), then the *Nc* is stored as a new case of the cluster represented by ρ.

The learning of new cases and the building of its representative prototype it is one of the aims to achieve in a DACL. To get a successful learning of cases and a good quality of learning *Mc's*, a Stochastic Learning *Mc's* Method "***SLMcM***" is introduced.

SLMcM considers two relevant moments to guide the learning. The **first moment** is described as:

$$\rho = \min_{j} arg\, D(Nc, Mc^{j}) \tag{1}$$

Formula 1 is computed to find the most similar *Mc* in DACL to the new case (*Nc). D* is the distance function that evaluates the dissimilarity value. In addition, ρ will represent the most similar prototype selected. The similarity of a new case will be assessed against all *Mc*'s by means of the same dissimilarity measure used in the normal assessment of similarity between two cases. In this case, the proposed measure is the *Euclidean distance* when all attributes are numerical, but other heterogeneous measures can be used when both numerical and categorical attributes exist.

The method for finding the most similar *Mc* can be summarized as follows:

```
Input: Nc (new case)
Output: $\rho$ (most similar Mc)

Dmin = + $\infty$
currentMc <- firstMc
while currentMc != null
  if(D(Nc,currentMc)<Dmin)then
    $\rho$ = currentMc;
    Dmin = D(Nc,currentMc)
  endif
  currentMc = nextMc
endwhile
return $\rho$
```

ρ value is the *Mc* most similar to the *Nc*. This *Mc* is the prototype that will store the new case (*Nc*) in the eventuality that the second condition will be true.

For the retrieval process, the same approach is implemented. When the retrieving of similar cases is executed, the first task is to find where to search. This can be done by finding the most similar *Mc*. Next task is the retrieving of similar cases. This has to be done following the indexing strategy of the sub-library. In our case, we have implemented the NIAR *k*-d tree jointly with a partial matching exploration technique (Orduña Cabrera, 2016), as previously explained.

The **second moment** to be checked is whether the following condition comes true or not:

$$\sigma(Nc) \le \sigma(\rho) * (1 + \gamma), \qquad \text{where } 0 \le \gamma \le 1 \tag{2}$$

γ is described as a virtual threshold added to the computed *Mc* threshold by $\sigma(\rho)$. The computation of $\sigma(Nc)$ and $\sigma(\rho)$ it is done considering all the attributes of *Nc* and *Mc* respectively. $\sigma(Nc)$ is computed by the summation notation of standard deviation formula. $\sigma(Nc)$ computes the deviation of *Nc* to the mean of the prototype ρ, and $\sigma(\rho)$ computes the standard deviation of the cases in the cluster represented by ρ.

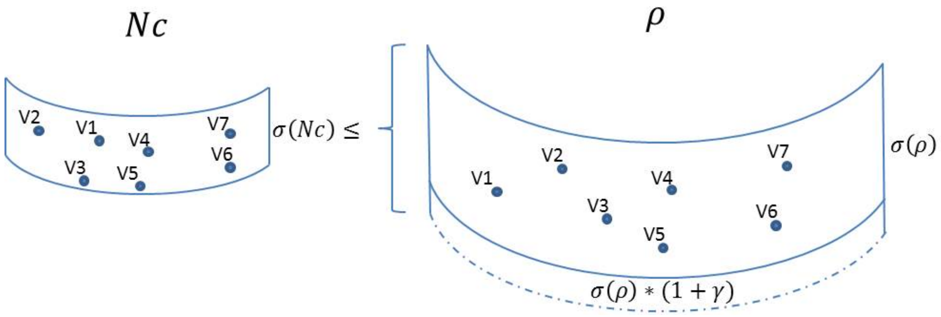


Fig. 2. Mc/ρ virtual threshold representation

Figure 2 represents a virtual size (normal line section) and resize (dot line section) of the *Mc* threshold; it has built in formula 2, when $\sigma(\rho) * (1 + \gamma)$ it is estimated. This threshold is used to decide whether a case is going to be stored or not in current *Mc*. The value $1 + \gamma$ indicates the percentage of the dispersion/deviation of ρ that will be considered as a *relaxation of the Mc value*. The value of γ can be less than 1, but, its adequate value has to be found. The implementation of the *relaxation process* considers endowing the *Mc* with a higher range of learning. The implementations of some evaluation tests have concluded that if the *relaxation process* is not implemented, the learning rate is reduced.

The standard deviation depends on data distribution and the state of the set of clusters depends both on the data distribution and on the previous states, being reached by the DACL.

If the *Nc* satisfies eq (1) and eq (2), it means that it belongs to the same data distribution (similar mean value and deviation, i.e, sub-concept) represented by the prototype ρ. On the other hand, if eq (2) is not satisfied, it means that it does not belong to the known data distributions until now (i.e., known sub-concepts) and that it probably is a data belonging to a new data distribution (i.e., new sub-concept)

2.3 The Stochastic Learning Policy

In the previous subsection, it has been introduced the details of the two core-DACL moments used in the learning policy algorithm. One of the open problems in clustering field is to select the number of clusters. This is relevant in DACL, too. With the following learning policy, a DACL is able to dynamically learn the number and content of the clusters and classify the continuous data precisely, according to the posterior experts' evaluation, as shown in the experimental evaluation section.

The aim of this policy is to learn those cases that accomplish the two moments of the policy (i.e., they belong to the same sub-concept) into already existent clusters. However, if they do not fulfill the two moments the cases should be learnt in a new formed cluster, because it probably belongs to a different data distribution, as previously detailed in last section. The data are stored in the sub-library that is most similar to the incoming case, according to the first moment in formula 1. If the case does not accomplish with the second moment (formula 2), a new sub-library is built. This policy has the following steps:

```
Begin policy
Var indValue      //tested $\gamma$ value
Nc arrives

//Find most similar to the incoming case according formula 1  $\rho = \min_j arg\, D(Nc, Mc^j)$
/* $\rho$ represents the sub-library where the Nc is stored and Mc is the prototype */

Var desvMc = $\sigma(\rho.Mc)$
Var desvNc = $\sigma(Nc)$
if(desvNc <= desvMc * (1 + indValue))
{
  StoreCases(Nc, $\rho$.Mc)
  Update($\rho$.Mc)
  return         //the process ends when the case is learned
}
else{
    BuildMc(Nc) /* a new Mc is created, the new Mc takes the values of the Nc*/
}
return
EndPolicy
```

At the beginning of the algorithm is defined the relaxation value for the prototypes. From the experimentation done, the best value of gamma relaxation of the dispersion parameter is 0.1 for the air quality dataset tested, according to the number of prototypes generated and their interpretation by the experts. Others values were tested but this threshold gave the best results (detailed in experimental evaluation section). In the step Update($\rho.Mc$), the value of the prototype is updated. This is done considering the values of the *Nc* saved in its structure. The update is done implementing the *Mc* building process, previously explained. When an arriving case does not accomplish the second moment, then a new prototype is created. This new prototype considers the values of the *Nc* as $\rho.Mc$ = *Nc* and store the case in its structure, initializing a new indexing structure (*k*-d tree).

3 A Case Study on Air Quality Assessment

An experimental evaluation of the method has been carried using a database of air quality of the city of Obregon, Sonora State, Mexico, for a period of one year. Obregón is one of the municipalities of the northwestern state of Sonora, Mexico. Its capital is Ciudad Obregón. The municipality has an area of 3,312.05 km² and a population of 784,342 inhabitants according to the National Institute of Statistics and Geography (INEGI) by its name in Spanish (INEGI 2010). The environmental condition aims to evaluate the air quality in the city, according to the international norms of the World Health Organization (WHO). The features considered are: PB (μg/m3), Temperature, Relative Humidity, RS (w/m2), PM10 (μg/m3), PM2.5 (μg/m3), Ozone (ppb), SO2 (ppb), NO (ppb), NO2 (ppb), NOX (ppb), CO (ppm). The evaluations were done each minute, following international norms.

In one day, different environmental conditions could happen. This actually depends on nature, but the human activities have a direct influence in the air quality. The people activities in the city are the industry, manufacturing, and the major activity, which is the agriculture. One fact that affects the air quality is the burning of sheaf at the end of seasons. The burning of sheaf has a direct influence on the seasonal behavior, where a concentration of pollutants is observed when the burning is done. Wheat cultivation is one of the main agricultural activities in the region of Cajeme. In the winter, the low levels of atmospheric pressure limit the dispersion of atmospheric pollutants. Other activity considered normal is the traffic at peak hours. These two activities are considered as the main cause of the air pollution in the city. For one year, the environmental features were monitored, and a database was created.

The aim of the set of tests is to evaluate the learning task of new prototypes for a dynamic adaptive case library (DACL). The Stochastic Method proposed was thought especially to guide the learning of the environmental conditions and storing the conditions as cases in a sub-library according to the proposed policies. The main motivation is to collaborate with environmental experts, providing a method that helps them to identify and evaluate the behavior of the air conditions, according to the norms of WHO. Nowadays, the experts evaluate the information using others tools, but in this process they need to take care of handling the information, especially when they elaborate reports and evaluate information to build conclusions. This task is complicated by the large amount of information generated in this domain. With proposed DACL policies, this process is dynamic, adaptive and automatic.

The analysis of a big amount of data is complicated to be done by the experts, which usually consider the use of spreadsheets. Therefore, the use of special analysis tools is required. For instance, the acquired information in one day is of 1440 information units (cases), considering the 12 attributes plus the time stamp attribute. These cases could be very different, because in one day different environmental conditions could happen. The natural conditions are modified by the human activities, and patterns of environmental behaviors could be found. One instance of this is the peak hours when people is going to pick up the children from school, or going to lunch at middle day, and finally, when day ends and people goes to house to rest. The last description of the human behavior could be found in the analysis of data. In addition, when a burning of sheaf happens, it could be found, too. To get conclusions of human activities related with air quality, the experts should split the information considering the time of each activity, and make inferences of the human affectation in air quality. The split of information with spreadsheets is easily done, considering time of activities, but when the split is done considering environmental conditions, the task is complex. The complication here was to find a method for identifying and organizing the stored cases in sub-libraries. This way, the environmental conditions were detected. According to the experts, the expected result of the method is that *should be easy to found the relation of human behavior and air quality*.

The stochastic method proposed is the tool developed to answer the requirement of the experts. The environmental experts working in the air quality evaluation require a computational method to organize the information acquired in the air quality assessment task. The method should behave such as the experts do to be fully trusted on. That means organizing the acquired data taking into account the human behavior in the urban areas. In our work, five human behaviors were considered for the evaluation of the prototypes obtained in the experimental work.

As the whole database with the information was very huge and complex, the case study was carried out in two steps:

- First, the experimentation was undertaken to analyse a smaller dataset containing the observations of one day with a 1-minute sampling period. That means 1440 observations. Due to some problem in the storing procedure of the database, one measure by each day was lost. Thus, from the 1440 measurements, one measure was lost and the finally analysed dataset had 1439 observations. This experimentation was done to preliminary test the approach and to better illustrate the idea of the dynamic process of discovering the air quality conditions in just one day for making it better understandable. However, a one-day record is not representative of the whole scenario.
- Thus, afterwards, we carried out the analysis of the whole dataset of 1-minute sampling of one year. From the 365 days, after a pre-processing step to avoid missing values and errors in data, finally there were 346 equivalent days. Thus, processing 497894 observations. As discussed in next section, the results on the complete one-year dataset were consistent with the previous analysis of just one day.

3.1 Evaluation Methods

Usually, an obtained partition (set of clusters) in a static clustering problem should be validated from two points of view: *structural validation* and *interpretation of the partition* obtained. From a structural validation perspective, what is usually done is checking for the compactness and separation properties of the clusters. Commonly, in the literature, this validation is done with some Cluster Validation Indexes (CVIs) (Sevilla-Villanueva *et al*., 2016), but other techniques can be used. These measures are structural validation measures about the *intra-cluster compactness* and *inter-cluster separation* dimensions.

Here, we have used other *structural validation techniques* described by several criteria expressed in two validation policies called *Policy One* (pol1) and *Policy Two* (pol2). They are complementary policies. They are activated in cascade. Policy 1 assess the *separation of the clusters*, while policy 2 assess the *compactness of the clusters*.

The policies are summarized as follows:

Policy One: The first policy is measuring the *separation of the clusters* through the computation of the distances among the prototypes (meta-cases). This is analogue to a separation CVI. The policy has the aim to measure the distance between *Mc*'s. The distance measure depicts the prototype distributions in DACL. If the prototypes are too close, the building of a new prototype merging the two overlapping prototypes might be considered to avoid the overlapping. Nevertheless, in the stochastic proposal the learning of a new case is obliged to comply with the two moments of the method described previously. This will ensure that the prototypes do not overlap, at the moment of learning a new case, but in next processing of new cases, two prototypes could overlap. The metric used to evaluate this dissimilarity is the Euclidean Distance if all attributes are numerical or some heterogeneous similarity measures (Gower similarity coefficient, etc.) to assess both the numerical and the categorical attribute dissimilarities.

Policy Two: With the second validation policy, which includes the computation of four formulas, we have estimated the *compactness of the clusters* (formulas 3 and 5), similarly to compactness CVIs, and the similarity of the prototypes (meta-cases) with the cases they represent (formulas 4 and 6). The policy is structured in four steps. To achieve that, the following formulas are computed:

$$Dav^j = \frac{1}{m}\left(\sum_{i=1}^{m} d(Mc^j, C^i)\right) \qquad (3) \qquad\qquad Mc^j = \sqrt{\sum_{k=1}^{n} \left(Mc_k^j\right)^2} \qquad (4)$$

$$SM^j = \sum_{i=1}^{m} d(Mc^j, C^i) \qquad (5) \qquad\qquad Mt^j = Mc^j / _{SM^j} \qquad (6)$$

Where n = # attributes describing the cases and metacases, and m = #cases(Mc^l)

Formula 3 computes an average of distances of all cases of the sub-library and its prototype (Dav). Dav depicts how compact is the sub-library. For a better quality of a DACL, the sub-libraries must be as compact as possible indicating that cases in the sub-library are similar enough to the prototype. In a DACL, a compact sub-library improves the learning rate. *Formula 4* computes the magnitude of the prototype. With the magnitude, the relation between the attributes of the *Mc* is evaluated. A higher value indicates that the pollution is higher, according to the environmental experts. With low values in attributes, better environmental conditions are met.

Formula 5 express the accumulation of the distances between the total of cases in the sub-library and the prototype. Where *j* indicates the current *Mc*, and *i* the current case. This measure is an indicator of how compact is the sub-library. *Formula 6* and the magnitude in formula 4 are used to estimate the similarity of the prototype with the cases that its represents. When the value of *formula 6* is low, the prototype and *Mc* have more in common. This particularity indicates that the attributes of the *Mc* are closely similar to the attributes of the cases in the sub-library. Therefore, a low value here is desired.

Regarding the *interpretation of the clusters*, the most used method is the evaluation and interpretation done by the experts/end-users. We have asked to the environmental experts in air quality to analyze the resulting clusters, and they have interpreted them as very meaningful clusters, because they are corresponding to the daily patterns of human activities. Here the interesting point is that these air quality conditions have been discovered automatically just from the data itself.

4 Discussion of Results

The implementation of the proposed method has been carried out using the database acquired in the year of evaluation in two steps, as described previously.

4.1 One-day results

The *Nc* arrives to the DACL. Then, following the stochastic steps that have to be accomplished, according to previous detailed algorithm, the *Nc* is processed. When the new case arrives (*Nc*), the first task is to evaluate the first moment of the proposal, which consists in finding the most similar prototype. When this prototype is found, the second moment consists of checking the learning value. In this step, the learning rate is evaluated and the experimentation is done with the experimental values. Finally, the decision to assign the new case to an existing prototype or to create a new prototype is made.

#Prototypes	**γ: Policy Evaluation Values**				
	0.1	***0.2***	***0.3***	***0.4***	***0.5***
1	472	695	883	1077	1320
2	305	594	556	362	119
3	295	150			
4	354				
5	13				

Table 1. Number of Prototypes, number of cases for each prototype and the different γ values for the one-day scenario.

Previously, in section 2, it has been introduced the core of the proposal; the stochastic method. At the beginning of the algorithm is defined the relaxation value for the prototypes.

The determination of the parameter gamma has been done in a trial and test experimental setting. Table 1 shows the different number of prototypes obtained, and the number of cases belonging to each prototype obtained according to the different values of the parameter gamma. Several values were tested (0.1, 0.2, 0.3, 0.4, and 0.5). We concluded that the best value for the parameter was $\gamma = 0.1$, after the expert's interpretation of the resulting number of prototypes and after the structural validation of the clusters obtained. The results with $\gamma=0.1$ obtained more prototypes, which after an accurate analysis matched better with corresponding air quality conditions. The second best value would be when $\gamma=0.2$. However, when γ is evaluated with others values, the results are out of expected bounds. Each different gamma value produces a possibly different number of final prototypes, and a different composition of each cluster corresponding to each prototype.

Table 1 shows the results of the implementation of the stochastic method. These results are from one day of acquiring of data. The behavior of more days of analysis was expected to be adding data to the current prototypes, as carried out in the second experimental testing. The comments of the environmental experts about the results obtained in the evaluation are the following.

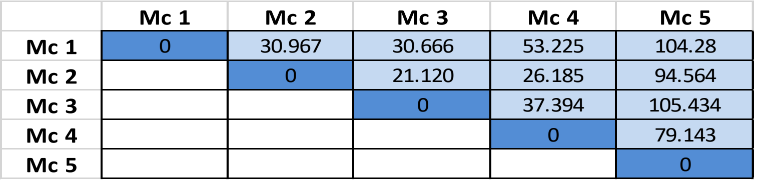
The five prototypes generated with $\gamma=0.1$ are *highly representative of the environmental conditions* taking into account the human activities. The *first prototype* built has 472 cases, where the first case is acquired at minute 00:01 and the last case learned is acquired about the minute 03:08. Between these minutes, usually the human activity is reduced to be sleeping, and the movement of cars is low. Usually, the traffic increase when the people decide to go schools. Between 3:00 and 7:00 hours, some initial activity is starting and the levels of pollution start to increase (*second prototype*). Normally, the children school starts at 07:30 hrs. Therefore, a peak hour is found between 07:00 and before of 13:00, because the pick-up from the school of the little children starts at 12:30 hrs. According to the table, the *third prototype* represents this human behavior. *Fourth prototype* matches the thirst human activity that starts around 13:00 hours ending about at 18:00 hrs. At this period, the people usually are at work. After this, between the time of 18:00 and 23:59 (*fifth prototype*), were the people travel to home or maybe is going to other places, and when the environmental condition represents the accumulative values of pollution of the whole day. After that, a new cycle starts and the nature has a break and helps us cleaning the environment. This is done during the first hours of the new day (first prototype).

When γ=0.2 the algorithm generates three prototypes. In this situation, the prototypes represent the behavior of the environment in three phases. First is between 0-12 hours; this behavior covers the human activities of the first and second period, according the prototype when the γ=0.1. Between the 12-22 hours covers the behavior when people takes a break to lunch and goes to pick up children to school, and finally when people ends the working day. In the 22-24 hours, the people travel to home or maybe going to other places. Having in consideration this arguing, the second γ policy evaluation value could be acceptable, but is more interesting and realistic the first one according the argumentation of the environmental experts.

Anyway, when γ experiments the other values of 0.3, 0.4 and 0.5, the obtained results are out of a reasonable behavior. One explanation of this fact is that the evaluation threshold is extended too much, and the relaxation learning policy is higher, mixing different prototypes in only one mixed prototype, which does not represent a clear different environmental situation, but a merge of several conditions. While in first evaluation of γ=0.1, the relaxation is more demanding, and only cases that accomplish the stochastic moments are learned.

A normal behavior of nature could be described as follows. At the end of the day, when the human activity is reduced and considering all the night and the first minutes (00:01) of the new day, the nature have a break, and at this time, the nature activity is more effective. Then, the improving of the air quality is higher. When the human activity begins, the air quality starts to going down. At late hours, the concentration of contamination increases and is reduced until the human activity is reduced. Therefore, the cycle ends and begins again.

Once the prototypes have been built, the following task is to evaluate the ***first policy***. The policy aims to measure the separation between *Mc*'s. The following table shows the results in the evaluation of the prototypes when γ=0.1. The prototypes evaluated have been selected having in mind the evaluation of γ, and the comments of the environmental experts. The normalization of the distance between 0 and 1 is a normal task in data mining. However, in this special case, the distances are computed on its natural values, with the aim of following the international and Mexican norms, as it is depicted in table 2.

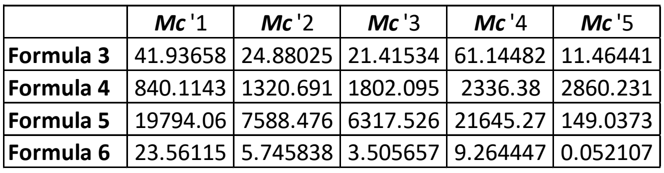


	Mc 1	Mc 2	Mc 3	Mc 4	Mc 5
Mc 1	0	30.967	30.666	53.225	104.28
Mc 2		0	21.120	26.185	94.564
Mc 3			0	37.394	105.434
Mc 4				0	79.143
Mc 5					0

Table 2. Distance measures between the *Mc*'s obtained for γ=0.1 in the one-day scenario

Table 2 shows the distance evaluation between prototypes. According to the results depicted in table, all prototypes are separated for a good distance between them. This is an indication that the prototypes are well structured, and the use of the stochastic steps works fine. As they are well separated, the probability that they overlap is low. However, to ensure that they do not overlap, an additional separation measure could be computed. This measure can be computed analyzing two clusters this way: for each case (x_i) belonging to one cluster check whether its distance to the prototype of the other cluster, $D(x_i, \rho_2)$, is higher than the relaxed deviation of the prototype of the other cluster ($\sigma(\rho_2)*(1+\gamma)$). If this property is satisfied for all the cases in the cluster, then the clusters do not overlap. On the other hand, according to the number of points failing the condition, a percentage of overlapping can be obtained. In fact, we assessed this property and there was some overlapping between prototype 1 and 2, and 1 and 3.

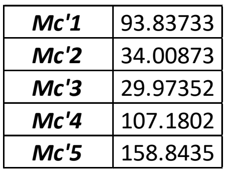
The second evaluation is the implementation of the ***second policy***. This is done through the implementation of the formulas 3, 4, 5 and 6 for the prototypes. Previously, it has been introduced the aim and meaning of the formulas, which is to evaluate the quality of the learning (or the building of new prototypes), starting with an empty case-base. The table 3 depicts the results of the computation of each formula for the prototypes.



	Mc '1	***Mc*** '2	***Mc*** '3	***Mc*** '4	***Mc*** '5
Formula 3	41.93658	24.88025	21.41534	61.14482	11.46441
Formula 4	840.1143	1320.691	1802.095	2336.38	2860.231
Formula 5	19794.06	7588.476	6317.526	21645.27	149.0373
Formula 6	23.56115	5.745838	3.505657	9.264447	0.052107

Table 3. Results of the different formulas assessment in the one-day scenario

Formula 3 computes an average while formula 5 computes the sum of distances between the cases and the prototype. These two formulas help to *view the compactness of the sub-library*. Especially in formula 5, it is possible to assess how far the cases to its prototype are. Here, a reduced value is desired, because it will be an indicator of a good compact sub-library (hard sub-library). A high value indicates that the cases are far from its prototype. Then, the class/cluster is not too compact. Thus, it is soft. Taking into account results in table 3, the *prototypes one and four* are the prototypes that could be soft, because both have the higher values, and where prototype 1 has more than *Mc* 4. Prototype 4 has the higher separation of its cases. These prototypes represents the first and last hours of the day, where the *human activity it is reduced*. *Prototypes two and three* are the harder prototypes, and are the prototypes which represents when the human activity is high. The harder prototypes are the cases more nearest to its prototype. The evaluation of the *magnitude of each prototype* is computed, and the results are depicted in table 3 as results of formula 4. According to the experts, in the morning the air quality is good, but with the human activity, the air quality is going worst. This behavior is expected in data. In the evaluation of the prototype, could be seen how the values for each prototype increases. The increase of magnitude shows us that the pollution has been increased. Finally, in formula 6, a similarity of the prototype and the evaluation of distances are computed. This similarity gives an idea of *how representative is the prototype of all the cases that are stored/summarized on it*. To have a full evaluation of the results the following table 4 shows the evaluation of the dispersion of the data in the prototypes. Table 3 in formula 3, 5 and 6 shows a pattern as result of the implementation of the stochastic method. Table 4 shows the standard deviation evaluation, which depicts the hard prototypes and soft prototypes. In hard prototypes, the cases tend to be very close to the ***Mc***, which is the situation of ***Mc'2*** and ***Mc'3***. Others prototypes are spread out over a large range of values. This behavior is depicted in both tables.



Mc'1	93.83733
Mc'2	34.00873
Mc'3	29.97352
Mc'4	107.1802
Mc'5	158.8435

Table 4. Standard Deviation of Prototypes

Table 3 and 4 complements the evaluation of the prototypes, but especially in the evaluation of prototype 5 the result is low, the magnitude is high, and the distance is low, but the number of cases is reduced. In this case, is essential the evaluation of the dispersion results that is the higher value and magnitude is higher too. It is highly interesting that the discovered patterns of air quality in the city matches with the citizen daily behavior.

4.2 One-year results

In the one-year experimentation, first the whole dataset with the information of one year was pre-processed. Several data errors and missing information in some days caused the removing of some data information (cases). Finally, 346 days were available for processing. This means 11.5 months of data, and 497894 observations/cases (346 * 1439).

Prototype	gamma:Dispersion values					
	0.075	0.085	0.1	0.15	0.2	0.3
1	165187	169411	174893	189487	204399	268308
2	40592	56770	119934	178199	213047	197728
3	101796	107681	102115	93435	66619	27742
4	83964	86521	58438	28972	10189	1537
5	51649	46098	30349	6214	2508	1841
6	30117	19753	8402	459	225	7
7	14439	7759	3435	906	773	713
8	5166	3075	328	222	121	5
9	2497	337			13	13
10	1427	290				
11	585	199				
12	475					

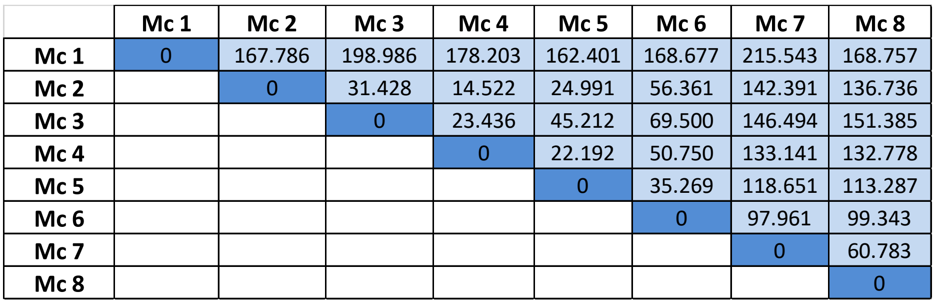
Table 5. Number of Prototypes, number of cases for each prototype and the different γ values for the 346 days scenario.

As in the previous experiments, the data stream was processed with different values of the γ parameter (dispersion of the deviation regarding the prototypes), to find the best value through this trial and error process. The γ values tested were 0.075, 0.085, 0.1, 0.15, 0.2 and 0.3. The results obtained are depicted in the table 4.

From an *interpretation point of view*, the behaviour of the parameter γ regarding the number of prototypes and the number of cases in the prototypes is equivalent to the previous one-day scenario. As the value of γ increases, the number of prototypes tend to decrease, and the number of cases in the earlier formed prototypes (i.e., those formed before than the others) tend to increase, making the prototypes more robust. Although with γ=0.2 and γ=0.3 the number of prototypes increase to 9, the number of cases of some of these new added prototypes is very low, especially compared with the other prototypes. Thus, probably, those prototypes are not very representative. After an interpretative analysis from the experts, they selected the value of γ=0.1 as the most interesting value, according the experimentation done, regarding an interpretation point of view. With γ=0.1, there were 8 prototypes. The first five prototypes are equivalent to the five prototypes found in the one-day scenario regarding the time and pollution levels. The *first prototype* is mostly formed by the hours between 0:00-3:00, with very good air quality condition, when human activity is mostly stopped. The *second prototype* is describing the hours between 3:00-7:00, when the air condition is good, but the pollution levels are starting start to increase due to the beginning of human activity. The *third prototype* is representing the hours from 7:00-13:00 where there is a lot of human activity (traffic, etc.) and with high levels of pollution. The *fourth prototype* includes the hours between 13:00-18:00, when people are usually at work and industries are fully operating, with high levels of pollution. The *fifth prototype* includes the hours between 18:00-23:59, when people is coming back to home, etc., with a lot of traffic generating high levels of pollution, accumulated during all the day.

The other 3 prototypes (6, 7 and 8) are very special ones. Making an accurate analysis of the number of cases of each one of the prototypes, it can be seen that the *number of cases in the prototypes 6, 7 and 8* (8402, 3435 and 328) is just the 2.4% of the cases, and on the other hand, *the first five prototypes* are covering the 97.6% of the cases. This means that the prototypes 6, 7 and 8 are very occasional. In fact, in an accurate analysis done, it was shown that the cases belonging to these rare prototypes were accumulated mainly in three weeks during the whole year (in April, September and November matching the spring and autumn agricultural activities), and in all other days of the year, there are no cases belonging to these prototypes. Moreover, analysing the time of these cases, most are between hours 17:00-19:00 in prototype 6, between 19:00-21:00 in prototype 7, and between 21:00-23:59 in prototype 8. Prototype 6 had higher levels of pollution than prototype 5. The interpretation from experts was that as these activities are restricted to some weeks in a year, and according to the time, probably could represent agricultural activities of burning of sheaf, some of which are forbidden, but some people do, in prototype 6. Prototypes 7 and 8 are variations of prototype 4, with similar pollution levels but in different time zones.

From the point of view of the *structural validation and definition of the prototypes*, we followed the same validation policies. Applying the *first policy*, the separation between *Mc*'s was measured. The following Table 6 shows the results in the evaluation of the prototypes when γ=0.1.

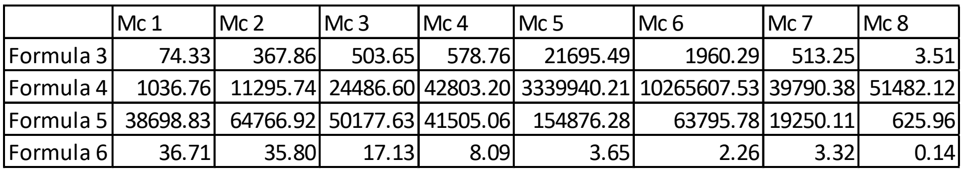


	Mc 1	Mc 2	Mc 3	Mc 4	Mc 5	Mc 6	Mc 7	Mc 8
Mc 1	0	167.786	198.986	178.203	162.401	168.677	215.543	168.757
Mc 2		0	31.428	14.522	24.991	56.361	142.391	136.736
Mc 3			0	23.436	45.212	69.500	146.494	151.385
Mc 4				0	22.192	50.750	133.141	132.778
Mc 5					0	35.269	118.651	113.287
Mc 6						0	97.961	99.343
Mc 7							0	60.783
Mc 8								0

Table 6. Distance measures between the MC's obtained for γ=0.1, in the one-year scenario.

Table 6 shows the distance evaluation between prototypes. According to the results depicted in table, all prototypes are separated for a reasonable distance between them. Especially, prototypes 6, 7 and 8 are more separated from the other five ones and among them. This is an indication that the prototypes are well structured, and the use of the proposed stochastic learning approach is reasonable. As they are well separated, the probability that they overlap is low. However, to be completely sure that they do not overlap, the same measure than in the previous one-day scenario to detect some overlapping was applied. There was some overlapping between prototypes 2 and 4, between 3 and 4, between 4 and 5, and between 5 and 6.

The second structural evaluation step is the implementation of the *second policy*. This is done through the implementation of the formulas 3, 4, 5 and 6 for the prototypes. The table 7 depicts the results of the computation of each formula for the prototypes.



	Mc 1	Mc 2	Mc 3	Mc 4	Mc 5	Mc 6	Mc 7	Mc 8
Formula 3	74.33	367.86	503.65	578.76	21695.49	1960.29	513.25	3.51
Formula 4	1036.76	11295.74	24486.60	42803.20	3339940.21	10265607.53	39790.38	51482.12
Formula 5	38698.83	64766.92	50177.63	41505.06	154876.28	63795.78	19250.11	625.96
Formula 6	36.71	35.80	17.13	8.09	3.65	2.26	3.32	0.14

Table 7. Results of the different formulas assesment in the one-year scenario.

Formulas 3 (distance average) and formula 5 (sum of all distances) among the cases of one prototype is evaluating the compactness of the prototypes. Formula 4 computes the magnitude of the prototype, and higher values represents higher levels of pollution. Formula 6 computes the degree of representation of each prototype regarding the cases in the prototype.

Regarding the *compactness of the prototypes*, through formulas 3 and 5, it can be seen that the prototypes are well defined, because they have rather low values excepting prototypes 5 and 6, which were overlapping a bit. Even that, they got these higher values in formula 3 due to the fact that they have less cases in the prototype, and the average distance is higher. Taking into account the *degree of representation of the prototypes*, the formula 6 outlines that prototypes 1 and 2 are not as highly representative as the other prototypes. The rare prototypes (6, 7 and 8) are highly representative of their cases. Formula 4 gives the *magnitude of the prototype*, which is correlated with the pollution levels of the cases represented by the prototype. Similarly to the one-day scenario, the values of the magnitude increase as the time is progressing from the morning (high quality) to the evening (worse quality). This can be seen through the different prototypes, from 1 to 5. In the *rare prototypes* 7 and 8, which are somehow some variations of the prototype 4, the levels of pollution are a little bit lower or higher than in prototype 4, but in other time zones. Prototype 6 has three times more pollution levels than prototype 5, probably because of the additional agricultural activities carried out in those weeks.

The observed patterns and the end of processing the whole year (346 days available) were patterns that were repeated along the whole year. The five prototypes find out in the one-day experimentation match with the first five prototypes found in the one-year scenario, and the additional prototypes discovered in the second scenario correspond to very special behaviour regarding agricultural activities in especial weeks of the year or variations of other prototypes regarding different time zones. Moreover, the value of the parameter γ =0.1 showed to be a good setting in both scenarios.

As explained above, the patterns discovered automatically through the data stream processing, compared with the actual data, were matching human/agricultural activities as the experts signalled. The opinion of the experts was that the methodology was very promising to detect, in an automatic way, different air quality conditions.

Therefore, it can be concluded that the utility of our approach for managing the identification of air quality environmental conditions from a data stream seems to be good.

Regarding the *prediction ability of the approach*, it was not described in this experimentation, because we focused on the identification problem (unsupervised databases) rather than the prediction (class label in supervised databases). However, in the application of the methodology to other datasets, we found that the accuracy of the prediction values of a class label, with this incremental prediction (dynamical DACL on-line predictor) were just between 5-7% lower than the use of the baseline non-incremental prediction technique (flat memory batch predictor) (Orduña Cabrera, 2016).

5 Further Experimentation

The main experimentation has been focused on the air quality conditions detection problem, which was quite complex. However, in order to explore the generalization abilities of our approach, it has been tested in other datasets, which has confirmed the suitability of the approach. Next, there are the description of those experiments.

			Number of Prototypes/Meta-cases and number of cases of each prototype						
DB	**#Classes**	**#Inst**	**γ=0.9**	**γ=0.425**	**γ=0.74**	**γ=1**			
Pima	2	768	Mc1=753, Mc2=15						
Iris	3	150		Mc1=50, Mc2=68, Mc3=32					
Ecoli	8	336			Mc1=144, Mc2=10, Mc3=102, Mc4=1, Mc5=21, Mc6=46, Mc7=2, Mc8=10				
Abalone	29	4177				Mc1=132, Mc2=38, Mc3=4, Mc4=63, Mc5=173, Mc6=16, Mc7=9, Mc8=26	Mc9=31, Mc10=100, Mc11=101, Mc12=33, Mc13=73, Mc14=85, Mc15=8, Mc16=78	Mc17=72, Mc18=24, Mc19=72, Mc20=228, Mc21=64, Mc22=42, Mc23=43, Mc24=20	Mc25=91, Mc26=82, Mc27=55, Mc28=1, Mc29=139, Mc30=151, Mc31=1, Mc32=1361, Mc33=761

Table 8. Number of prototypes and number of cases for the Pima, Iris, Ecoli and Abalone datasets, according to the best γ value obtained.

Several datasets from the UCI-ML repository (Lichman, 2013) were used. In table 8, there is a summary of the results of applying our approach in four datasets: Pima, Iris, Ecoli and Abalone. The table depicts the number of classes, the number of instances, and the number of prototypes and the cases within the prototypes according to the different values of parameter γ. Of course, as these datasets were supervised, we hide the class label for the experimentation of processing the whole data stream of each dataset. As shown in the table 8, in some datasets (Pima, Iris and Ecoli) the real number of prototypes could be discovered, and the cases represented by the prototype were matching the actual cases in the dataset.

For instance, in the Iris dataset, the three classes were discovered, and the 50 examples of the class Iris-setosa were completely identified. The other ones were a little bit confused, but not more than usual static clustering algorithms.

In the case of Abalone dataset, our approach discovered 33 different prototypes, when there are 29 different classes. Notwithstanding, the problem is very difficult because 5 classes have just one example, and two classes have just two examples (imbalanced dataset). Thus, these 7 classes are very difficult to discover.

Nevertheless, it was observed that even the number of discovered prototypes was not exactly identified the *predictive accuracy was even greater* than identifying the exact number of prototypes. These probably means that some overlapping in the prototypes is not necessarily bad, especially for a predictive task.

In general, we can say that from the additional experiments done with other datasets, the generalization ability of our approach seems to be good.

6 Conclusions and future work

A stochastic learning method has been introduced to provide an efficient data stream mining strategy in the whole data science process in environmental data streams. The aim of the proposal is automatically to discover different prototypes, which corresponds to different sub-concepts hidden in the data. These sub-concepts are different states of a stochastic process handled by the data stream. The approach starts with an empty DACL (none cases and none prototypes) and the procedure is to dynamically process and learn the new cases and the new prototypes, and store them in the DACL structure being constructed. When current prototypes do not accomplish with the requirements of the method to learn the new case, then a new prototype must be created. This is done considering the new case such as the new prototype. With this proposal, a DACL is able to learn new cases and build new prototypes. The learning of new cases is done according to the most similar prototype (first statistical moment of the stochastic process, i.e., the mean). In addition, the second statistical moment of the stochastic process (i.e., the standard deviation) must fulfill the relaxation condition, in order that the new case must be stored in the most similar prototype. This way, we are proposing an effective framework for managing continuous domains or data streams, using case-based reasoning in environmental domains.

The proposed approach has been validated in a case study on air quality assessment, from a data stream database of air quality of the city of Obregon, Sonora State, Mexico, for a period of one day in a first scenario and one year in a second scenario. The number and content of the several prototypes has been achieved according to the different values of the main parameter of the stochastic learning method (γ parameter). With the value γ=0.1, five prototypes were discovered in the one-day scenario, which match with five typical environmental conditions closely related with typical human behavior. In the one-year scenario, eight prototypes were discovered with a value of γ=0.1, but the first five prototypes were matching the same prototypes obtained in the one-day scenario. The remaining three prototypes are rare prototypes matching special conditions in some weeks along the year, and being some variants of other prototypes at different time zones. The discovered patterns were evaluated according to several structural validation criteria, like separation among the prototypes, compactness inside the prototypes, and the representativeness of the prototype. They showed to be quite compact, quite separated, even some overlapping percentage existed among some of them, and enough representative of the cases belonging to them, according to the evaluation measures based in the four formulas described. In addition, the inspection of the characterization of the prototypes, like the hour times involved in the prototypes, and the levels of pollution gave an additional interpretative reason to validate the prototypes obtained.

In the opinion of the environmental experts, and from the results of experimentation, the method proposed here can be considered trustworthy for identifying and organizing the cases of air pollution conditions. In this approach, a DACL is able to learn new cases and build new prototypes in an incremental and automatic way. This way, we are proposing an effective framework for mining continuous environmental data streams, using dynamic case-based reasoning and providing the air quality experts with reliable knowledge for decision-making in environmental management.

The generalization ability of our approach was tested with other datasets, from UCI repository, with good identification skills as showed in the previous section.

In addition, the predictive capabilities of the approach, which were not the focus of this paper, was tested comparing the ability of our dynamic approach for making predictions in some datasets against using a static approach, with a flat memory scheme, and the accuracy of the predictions just dropped a 5-7%.

In future work, other different strategies for the case learning and meta-case building can be designed and tested in environmental data streams. Moreover, a strategy for determining automatically and adaptively the value of the γ parameter will be studied. An initial idea can be to start the process of the data stream with the highest value of γ=1, and depending on the predictive accuracy obtained on some cases, the value of parameter γ would be adaptively decreased by a constant value (i.e., 0.05). In addition, several other strategies regarding the possibility of removing or fusing meta-cases, in the dynamic process of the data stream mining will be analysed.

Acknowledgements. Authors express their acknowledgement for the support of Dr. Alberto Ramírez Treviño and Dr. Martín Villa-Ibarra from ITESCA (Instituto Tecnológico Superior de Cajeme) and from Environmental

Engineering Dept. at ITESCA and from Spanish Network DAMA (TIN2015-70308-REDT, MINECO /FEDER EU).

References

Aggarwal, C. Data Streams: Models and Algorithms. Springer, 2007.

Babcock B., Babu S., Datar M., Motwani R., and Widom J. Models and issues in data stream systems. In Proceedings of the 21st Symposium on Principles of Database Systems, pages 1–16. ACM Press, 2002.

Carslaw David C., Ropkins Karl. Openair- An R package for air quality data analysis, *Environmental Modelling and Software* 27-28, 52-61, 2012.

Costabile F. and Allegrini I. A new approach to link transport emissions and air quality: An intelligent transport system based on the control of traffic air pollution. Environmental Modelling and Software 23, 258-267, 2007.

De Coensel B., Can A., Degraeuwe B., De Vlierger I., Botteldooren D., Effects of traffic signal coordination on noise and air pollutant emissions. Environmental Modelling and Software 35, 74-83, 2012.

Fallah-Shorshani Masoud, André Michel, Bonhomme Céline, Seigneur Christian. Modelling chain for the effect of road traffic on air and water quality: Techniques, current status and future prospects. *Environmental Modelling and Software* 64, 102-123, 2015.

Finestrali, Giulio and Muñoz-Avila, Héctor (2013). Case-Based Learning of Applicability Conditions for Stochastic Explanations. Case-Based Reasoning Research and Development, volume 7969, pages 89-103.

Gama, J. and Mohamed Medhat Gaber (Eds.). “Learning from Data Streams: Processing Techniques in Sensor Networks”, Springer, 2007.

Gama, J. Knowledge Discovery from Data Streams. Chapman and Hall/CRC, 2010.

Gulliver John, Morley David, Vienneau Danielle, Fabbri Federico, Bell Margaret, Goodman Paul, Beevers Sean, Dajnak David, J-Kelly Frank, Fecht Daniela. Development of an open-source road traffic noise model for exposure assessment. *Environmental Modelling and Software* 74, 183-193, 2015.

Halonen, J.I., Marta Blangiardo, Mireille B. Toledano, Daniela Fetch, John Gulliver, H. Reggente Matteo, Peters Jan, Theunis Jan, Van Poppel Martine, Rademaker Michael, Kumar Prashant, De Beats Bernard, Prediction of ultrafine particle number concentration in urban environments by means of Gaussian process regression based on measurements of oxides of nitrogen, *Environmental Modelling and Software* 61, 135-150, 2014.

Haris S. and R. Slobodan (2005). Autonomous Creation of New Situation Cases in Structured Continuous Domains. Springer-Verlag, pp. 537—551.

Hill David J., S. Minsker Barbara, Anomaly detection in streaming environmental sensor data: A data-driven modeling approach, *Environmental Modelling and Software* 25,9, 2010.

Hulten, G. and P. Domingos. Catching up with the data: research issues in mining data streams. *In Proc. of Workshop on Research issues in Data Mining and Knowledge Discovery, 2001.*

Hulten, G., L. Spencer, and P. Domingos. “Mining time-changing data streams”. In F. Provost, editor, Proceedings of the *Seventh International Conference on Knowledge Discovery and Data Mining*. ACM Press, 2001

INEGI, 2010. Census of Population and Housing (in Spanish: Censo de Población y Vivienda). Last access in February 2018. http://www.inegi.org.mx/est/contenidos/proyectos/accesomicrodatos/cpv2010/default.aspx

Keane M. T. and B. Smyth (1995). “Remembering to Forget: A Competence-Preserving Case Deletion Policy for Case-based Reasoning systems”. Procc. of *IJCAI 1995*, Morgan Kaufmann, 377-382.

Klinkenberg, R. and T. Joachims. Detecting concept drift with support vector machines. In P. Langley, editor, Proceedings of *ICML-00*, pages 487–494. Morgan Kaufmann Publishers, San Francisco, US, 2000.

Klinkenberg, R.. “Learning drifting concepts: Example selection vs. example weighting”. *Intelligent Data Analysis*, 8(3):281 – 300, 2004.

Klinkenberg, R. and I. Renz. “Adaptive information filtering: Learning in the presence of concept drifts”. In Learning for Text Categorization, pages 33–40. AAAI Press., 1998

Kruusmaa M. (2003). Global Navigation in Dynamic Environments Using Case-Based Reasoning. *Autonomous Robots* 14, pp. 71-91.

Kubat, M. and G. Widmer. Adapting to drift in continuous domain. In Proceedings of the 8th *European Conference on Machine Learning*, pages 307–310. Spinger Verlag, 1995.

López de Mántaras, D. McSherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M. L. Maher, and M. T. Cox, K. Forbus, M. Keane, A. Aamodt and I. Watson. “Retrieval, reuse, revision and retention in case-based reasoning”. *The Knowledge Engineering Revie*w 20(3), 215-244, 2005.

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Liu Yi, Hu Jiameng, Snell-Feikema Isaiah, S. VanBemmel Michael, Lamsal Aashis, C.Wimberly Michael, Software to facilitate remote sensing data access for disease early warning systems, *Environmental Modelling and Software* 74, 247-257, 2015.

Maloof, M. and R. Michalski. Selecting examples for partial memory learning. *Machine Learning*, 41:27–52, 2000

Meléndez J, J. Colomer and J. Ll. de la Rosa. “Expert Supervision Based on Cases”. Proc. *of 8th IEEE International Conference on Emerging Technologies and Factory Automation*, Vol. 1, pp. 431—440, 2001.

Miyashita K. and K. Sycara. Improving system performance in case-based iterative optimization through knowledge filtering. Procc. of *IJCAI 1995,* Morgan Kaufmann, pp. 371—376. 1995.

Orduña Cabrera, F. A Dynamic Adaptive Framework for improving Case-Based Reasoning System Performance. Ph.D. Thesis. Universitat Politècnica de Catalunya, 2016.

Orduña Cabrera, F. and M. Sànchez-Marrè, Using NIAR-trees to Improve the Case-Based Reasoning Retrieval Step. 12th Mexican International Conference on Artificial Intelligence (MICAI 2013). Proceedings Part II. Springer Verlag, LNAI, vol. 8266. pp. 314-325. ISBN 978-3-642-45110-2.

Orduña Cabrera, F. and M. Sànchez-Marrè, M. Dynamic Adaptive Case Library for Continuous Domains. In Proc. of 12th International Conference of the Catalan Association of Artificial Intelligence (CCIA'2009). Frontiers in Artificial Intelligence and Applications Series, Vol. 202, pp. 157-166, 2009.

Ram A., R. C.Arkin, K. Moorman and R. J. Clark (1997). Case-based reactive navigation: a method for on-line selection and adaptation of reactive robotic control parameters. *IEEE System, Man, and Cybernetics Part B* 3, pp. 376-394.

Ram A. and J. C. Santamaría (1997). “Continuous Case-Based Reasoning. *Artificial Intelligence* 90, pp. 86-93.

Reis Stefan, Seto Edmund, Northcross Amanda, W.T. Quin Nigel, Convertino Matteo, L. Jones Rod, R. Maier Holger, Schlink Uwe, Steinle Susanne, Vieno Massimo, C. Wimberly Michael, Integrating modelling and smart sensor for environmental and human health, *Environmental Modelling and Software* 74, 238-246, 2015.

Richter, M.M. and R.O. Weber. “Case-Based Reasoning: a text-book”. Springer-Verlag, 2013.

Sànchez, M., Cortés, U., R.-Roda, I. and Poch, M. Using Meta-cases to Improve Accuracy in Hierarchical Case Retrieval. *Computación y Sistemas* 4(1):53-63, 2000.

Sànchez-Marrè M., U. Cortés, I. Rodríguez-Roda, and M. Poch. “Sustainable case learning for continuous domains”. *Environmental Modelling and Software* 14(5):349-357, 1999.

Sànchez, M.,Cortés, U., and Béjar, J., De Grácia, J., Lafuente, J. and Poch, M. Concept Formation in WWTP by Means of Classification Techniques: A Compared Study. Applied Intelligence 7(2):147-166, 1997.

Sevilla-Villanueva, B., K. Gibert and M. Sànchez-Marrè (2016). Using CVI for Understanding Class Topology in Unsupervised Scenarios. 17th Conference of the Spanish Association for Artificial Intelligence (CAEPIA 2016) LNAI, Vol. 9868, pp. 135-149. Springer-Verlag, 2016.

Tan, Swee Chuan and Ting, KaiMing and Teng, ShyhWei. A Comparative Study of a Practical Stochastic Clustering Method with Traditional Methods. AI 2010: Advances in Artificial Intelligence, vol. 6464, pp. 112-121.

Tang U.W, Wang Z.S, Influences of urban forms on traffic-induced noise and air pollution: results from a modelling system. *Environmental Modelling and Software* 22, 1750-1764, 2007.

Thaker Prashant and Gokhale Sharad. The impact of traffic-flow patterns on air quality in urban street canyons. *Environmental Pollution* 208, 161-169, 2016.

Urdiales C., E.J. Pérez, J. Vázquez-Salceda, M. Sànchez-Marrè and F. Sandoval (2006). “A Purely Reactive Navigation Scheme for Dynamic Environments using Case-Based Reasoning”. *Autonomous Robots* 21, pp. 65-78.

Vedrenne Michel, Borge Rafael, Lumbreras Julio, Encarnación-Rodríguez María. Advancements in the design and validation of an air pollution integrated assessment model for Spain. *Environmental Modelling and Software* 57, 177-191, 2014.

Widmer, G. and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23:69–101, 1996.

Wyat Appel K., C.Gilliam Robert, Davis Neil, Zubrow Alexis, C.Howard Steven, Overview of the atmospheric model evaluation tool (AMET) v1.1 for evaluating meteorological and air quality models, *Environmental Modelling and Software* 26, 434-443, 2011.