

On-line Classification of Data Streams with Missing Values based on Reinforcement Learning ^{*}

M. Millán-Giraldo, V. Javier Traver, and J. Salvador Sánchez

Institute of New Imaging Technologies (<http://init.uji.es>),
Universitat Jaume I, 12071, Castellón (Spain)
{mmillan, vtraver, sánchez}@uji.es

Abstract. In some applications, data arrive sequentially and they are not available in batch form, what makes difficult the use of traditional classification systems. In addition, some attributes may lack due to some real-world conditions. For this problem, a number of decisions have to be made regarding how to proceed with the incomplete and unlabeled incoming objects, how to guess its missing attributes values, how to classify it, whether to include it in the training set, or when to ask for the class label to an expert. Unfortunately, no decision works well for all data sets. This data dependency motivates our formulation of the problem in terms of elements of reinforcement learning. The application of this learning paradigm for this problem is, to the best of our knowledge, novel. The empirical results are encouraging since the proposed framework behaves better and more generally than many strategies used isolatedly, and makes an efficient use of human effort (requests for the class label to an expert) and computer memory (the increase of size of the training set).

Keywords: Reinforcement learning; Active learning; Adaptive learning; Streaming data; Incomplete data; Imputation techniques; On-line classification

1 Introduction

In many streaming data applications, where objects arrive one at a time, data may come with one or more missing attributes. Usually, the lack of some attributes causes statistical distortions in the data, which might degrade the predictive model and possibly lead to considerable reductions in the classification accuracy. Conventional techniques of dealing with missing data usually solve this problem by ignoring the incomplete objects (*projection*) or filling the missing attributes with values estimated from complete objects using statistical measures (*imputation*). Several approaches have been proposed in the literature [1–3].

In general, these results show that the application of methods of handling missing attributes might be helpful to improving the classification accuracy of incomplete data. However, in our experience [4], no single method or strategy is generally suitable for all

^{*} This work has been supported in part by the Spanish Ministry of Education and Science under grants CSD2007–00018 (Consolider Ingenio 2010) and TIN2009–14205, and by Bancaixa under grant P1–1B2009–04.

the different data sets. Therefore, in this paper we developed a framework for solving the classification of incomplete data streams, which is inspired in part on reinforcement learning (RL) in order to combine different approaches of handling missing attributes. Besides, since in RL, the learning process is done by interacting with the environment, we developed the method with the intention that it could adapt to the peculiarities of each data set and even each object with respect to their missing values.

The theory of RL has been broadly used for applications of engineering control, robotics and planning. Few works of machine learning (ML) based on RL have been proposed in the literature; however, its interest has increased in the last decade, [5, 6]. Nevertheless, in spite of these few efforts to the best of our knowledge RL has not been explored in the classification of data with missing attributes. The reason behind of this may be that there is not an immediate formulation in RL under this learning context. In this paper, the motivation for the use of RL as a framework for classification of incomplete data, is to find a general solution which is adaptive to the classification problem so that it is independent of the peculiarities of the data set. In the proposed framework the algorithm, based on ideas from RL, learns a prediction model in the sense that it handles incomplete objects and classify them. For this purpose, the algorithm includes five techniques of handling missing attributes, which one of them is the projection approach and the remaining ones correspond to support vector regression, nearest neighbor, random and mean imputation methods. Additionally, we introduce an action which the algorithm may ask the expert for the class label of a given object.

2 Reinforcement learning

In this paper, the proposed algorithm is based partially on Q-learning and, more specifically, on *temporal difference learning* (TD). This method was chosen because it is a simple technique and its structure is adequate for solving general optimization problems of learning [7]. Besides, it does not require a priori model of the environment or for the actions selection, then it must learn an action-value function (Q-function) as it interacts with the environment. These fit very well to our problem, where the characteristics of each data set are completely unknown.

2.1 Formulating the problem of classification of incomplete data using RL

We propose an RL-based framework to handle the problem of data streams with incomplete and unlabeled objects [8]. In this work, each object arrives with only one missing attribute, being the most relevant. The attribute relevance is measured using the Jeffries-Matusita distance [9]. Here, states are defined for the objects and not for the data stream. This is not the single way to formulate the problem, since different states and actions definitions may lead to different solutions. In this work, each state represents the current state of the incoming object, that is, if the object is complete x or incomplete x^j (without the attribute j), and if it was classified, considering the obtained confidence in its classification. Four states were defined:

State 1. *Incomplete object:* the incoming object, x^j , has one missing attribute in j .

State 2. Complete object: the missing attribute of x^j has been filled with either imputation approach.

State 3. Classified complete object: x is classified with a confidence p higher or equal than a minimum predefined threshold, th_{\min} (more on this later).

State 4. Classified incomplete object: the incomplete object x^j is classified, using the projection method, with confidence $p \geq th_{\min}$.

Actions defined in the algorithm are applied to the incoming object. They are related basically with (1) the methods of dealing with incomplete data, (2) classification, (3) whether to incorporate or not the incoming object into the training set and (4) returning to the initial state. Nine actions were included in the algorithm:

Action 1. Projection: it classifies the incomplete object using the projection approach, which objects from the training set, T , are projected to one dimension less.

Action 2. SVR: the missing attribute j is imputed using support vector regression (SVR) [4]. To this end, objects from T are used to build a new space using SVR. Then, x^j is mapped to this space to estimate the value for the missing attribute.

Action 3. 1NN: the missing attribute of x^j is filled with the attribute j of its nearest neighbor (1NN).

Action 4. Random: the missing attribute of x^j is imputed with a random value estimated in the range between the minimum and maximum values of the attribute j from the training set objects.

Action 5. Mean: the missing attribute of x^j is filled with the mean value obtained from attributes j of objects from T .

Action 6. Expert intervention: the expert provides the class label of the incoming object when the algorithm asks for it. The aid of the expert may be requested either for complete or incomplete objects, but it is only for the true class label and never for the missing attribute.

Action 7. Classification: it classifies x using the confidence based on its k -NN [4].

Action 8. Insert x into T : objects classified with $th_{\min} \leq p \leq th_{\max}$, where th_{\max} is a maximum predefined threshold, are incorporated into T with the aim to insert only objects which may provide useful information.

Action 9. Return to the initial state: if in the classification $p < th_{\min}$ the algorithm returns to the initial state to repeat the process using a different method of dealing with missing attributes.

Actions may be applied depending on the state where the algorithm is. Allowed actions in each state are shown in Table 1. When the algorithm tries to perform any invalid action it is penalized. Additionally, whether the algorithm reaches the final goal, which is to classify the object with confidence $p \geq th_{\min}$, it is rewarded. Thus, the algorithm keeps adapting its Q-values as it goes learning. Details of parameter values, such as rewards and penalizations, were not here included due to the lack of space.

3 Experiments and results

In this section we describe the experiments carried out for evaluating empirically the performance of the algorithm. Experiments were conducted as follows:

Table 1. Allowed actions in each state.

State:	1	2	3	4
Actions:	1 to 6	6 and 7	8	9

Data sets: 18 real data sets were used in the experiment (see Table 2). We normalized the data sets in the range $[0, +1]$, in order to avoid the possible influence of the difference between attribute scales on the results.

Partitions: Data sets were split into two sets. One of them is for the training set with a size of $d \times c$, where d and c are the number of features and classes [10], respectively. The remaining objects are employed for the streaming data set. Objects for data stream were shuffled before each run with the aim to simulate an independent and identically-distributed sequence. Data partitions for the initial training and streaming data sets were the same for all methods, and objects were presented in the same order with the aim to avoid that the order of data affects in the comparison of performances.

Repetitions: Experiments were carried out ten times for all data sets, with different training and streaming data sets each time, this with the aim to have a general behavior for each data set.

Incomplete objects: An object with one missing attribute arrived to the system one at a time. The missing attribute chosen for experiments was the most relevant attribute, that means, the attribute with the most discriminative information.

Expert intervention: To prevent the algorithm from constantly asking the expert for a class label, a constraint is imposed so that at least n objects have to be processed before asking the expert again. In real applications n should be set according to the actual cost/benefit ratio. The arbitrary value $n = 7$ was used in these experiments.

Classification: It is performed according to the probabilities of its k -nearest neighbors, from the training set, of belonging to each class [4]. Objects classified with $th_{\min} \leq p \leq th_{\max}$ are incorporated into the training set with the aim to insert only objects which may provide useful information. Thresholds values were automatically estimated from training set objects for each run. The classification error is estimated as the number of misclassifications divided by the numbers of objects seen until the moment by the algorithm. Besides, the classification error was averaged over the 10 runs in order to have a single performance for each data set.

Table 2. Data sets used in the experiments.

Repository	Ref.	Data sets
UCI	[11]	iris, wine, sonar, thyroid, heart, liver, voice9, wbc, australian, pima, vowel, german
Ripley	[12]	crabs
Library	[13]	laryngeal1, intubation, spect, laryngeal2
Private		Images of pieces of kerogen extracted from microscope images of palynomorphs

3.1 Analysis of results

Results obtained with the proposed RL-based framework were compared with five techniques of dealing with incomplete data: projection, SVR, 1NN, random and mean imputations techniques. As the baseline, we employed results obtained by classifying the same incoming objects but with all their attributes, which is called *Complete*.

Table 3 shows the final classification error (i.e. the accumulated error after all objects in the stream arrived), for each method (columns) and for the 18 real data sets (rows). Error values highlighted in bold indicate the best final error obtained for each data set. Interestingly, the method based on RL outperforms the other techniques (including the baseline case which has no missing attribute!) in 16 out of the 18 data sets. This result points to the general good behavior of the algorithm: whereas the other methods work the best for *some* data sets but not for some others, the proposed algorithm works the best for *almost all* the data sets. On the other hand, the percentage of objects (last column) for which the expert provided the true class label is relatively low (around 4–11%, depending on the data set). This suggests the ability of the algorithm of being sparing with the expert knowledge while benefiting from it.

Table 3. Final classification error

Data set	Complete	RL	Projection	SVR	1NN	Random	Mean	Expert [%]
iris	0.209	0.150	0.277	0.251	0.231	0.350	0.344	6.23
wine	0.050	0.024	0.043	0.046	0.049	0.045	0.044	9.64
crabs	0.425	0.403	0.435	0.441	0.444	0.460	0.476	7.02
sonar	0.232	0.199	0.238	0.235	0.231	0.239	0.241	10.57
laryngeal1	0.193	0.162	0.193	0.194	0.193	0.211	0.192	<u>11.38</u>
thyroid	0.244	0.198	0.258	0.253	0.250	0.325	0.291	5.05
intubation	0.359	0.273	0.405	0.381	0.402	0.397	0.405	10.49
heart	0.465	0.415	0.464	0.461	0.467	0.463	0.463	9.37
liver	0.452	0.432	0.453	0.455	0.452	0.456	0.456	8.14
spect	0.291	0.251	0.302	0.293	0.298	0.283	0.305	8.05
voice9	0.629	0.603	0.627	0.630	0.629	0.659	0.636	6.89
wbc	0.057	0.038	0.061	0.060	0.061	0.073	0.072	7.35
palynomorphs	0.186	0.160	0.201	0.187	0.198	0.222	0.206	10.85
australian	0.165	0.142	0.166	0.166	0.166	0.166	0.166	10.98
laryngeal2	0.067	0.060	0.067	0.067	0.066	0.068	0.065	<u>3.80</u>
pima	0.336	0.357	0.345	0.346	0.347	0.353	0.349	10.09
vowel	0.444	0.478	0.490	0.471	0.477	0.599	0.559	6.52
german	0.378	0.343	0.395	0.395	0.408	0.401	0.400	10.48

To present an example of the results, we choose one case when the algorithm works well (iris data set) which represent the general behavior for the most data sets (16 out of 18) and another one which show the worst case (pima data set). As it can be seen in Fig. 1(a), the classification error for iris data set has a decreasing trend; which could suggest that the algorithm is learning which methods of handling missing attributes are more appropriate for this data set. Additionally, the algorithm is incorporating only few objects to the training set, Fig. 1(b), unlike what occurs with the other methods. The cause of the good behavior is probably due to that only those objects that contain useful information could be being inserted into the training set, and they could be being used in subsequent classifications.

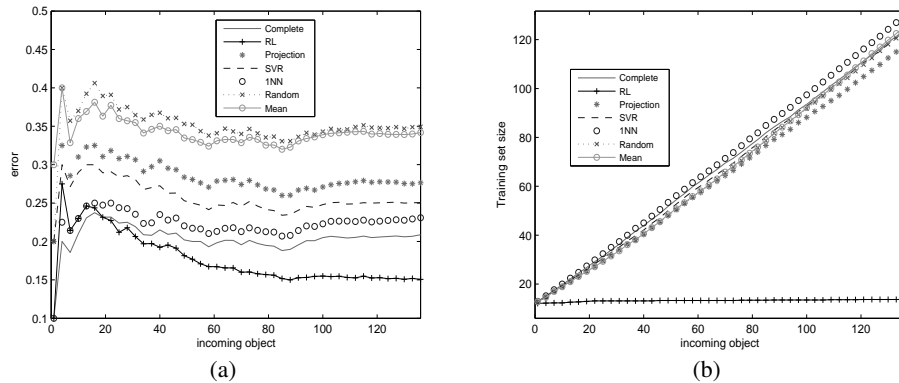


Fig. 1. Results for iris data set. (a) Classification error computed for ten runs, when the missing attribute was the most relevant. (b) Training set sizes of each technique.

The percentage of objects that used the actions of handling incomplete objects is presented in Table 4. As it can be seen, the technique more employed for iris data set was the 1NN imputation. On the other hand, in Fig. 1(a), among the projection and imputation methods, the 1NN is the curve with the best performance. Therefore, it may be that the algorithm has identified this technique as the most suitable for this data set. Probably, the use of the expert aid is influencing the results for iris data set. However, the percentage of objects that used the expert action was only about 6.23% (Table 3).

Table 4. Percentage of objects that employs each action of handling incomplete data.

	Projection	SVR	1NN	Random	Mean
iris	2.75	16.96	55.00	10.58	14.57
pima	28.21	20.92	19.60	16.28	14.91

For the pima data set, the best performance is obtained by projecting the objects (Fig. 2(a)). However, by design, objects which are projected are decided not to be included in the training set (just because they are incomplete). As a result, even if the true class label is provided by the expert, this information cannot be exploited for the classification of subsequent objects, which may explain the poor performance of the system in this scenario. Therefore, this design issue should be reconsidered in our future work.

A controlled experiment is carried out with the aim to verify that the RL algorithm used the expert action when it is supposed to, i.e. when an object is harder to classify. For that purpose, we sort the objects by difficulty of classification before they are inserted into the system. To make such a sorting, we organized them according to their posterior probabilities obtained by classifying them using the remaining objects and the confidence of their 5 nearest neighbors. Figure 3 shows, for spect data set, how much the expert was asked for the class label for the spect data set. In this figure, input objects

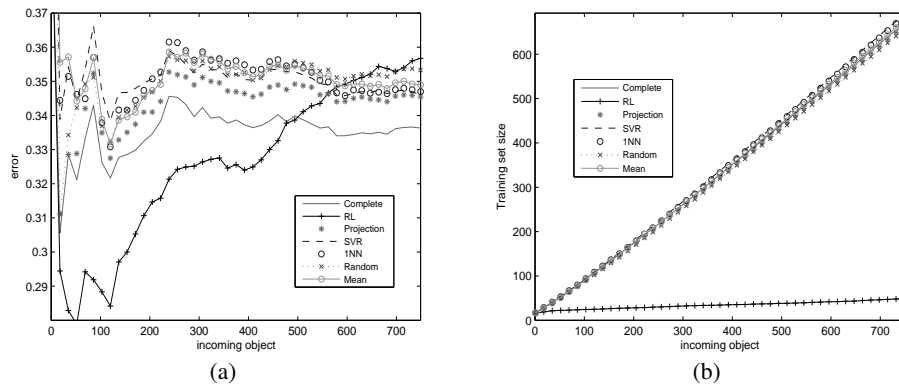


Fig. 2. Results for pima data set. (a) Classification error computed for ten runs, when the missing attribute was the most relevant. (b) Training set sizes of each technique.

were grouped in ten blocks by intervals of time. Figure 3(a) shows the results when the easiest objects for classifying arrived first and then the hardest, (b) otherwise. As can be seen in Fig. 3(a), the percentage of objects that make use of the expert action is increasing, whereas in Fig. 3(b) this percentage is decreasing. This is because the more difficult to classify objects are the more the algorithm requests the expert, and vice versa. However, this interesting behavior is not observed for most of the other data sets.

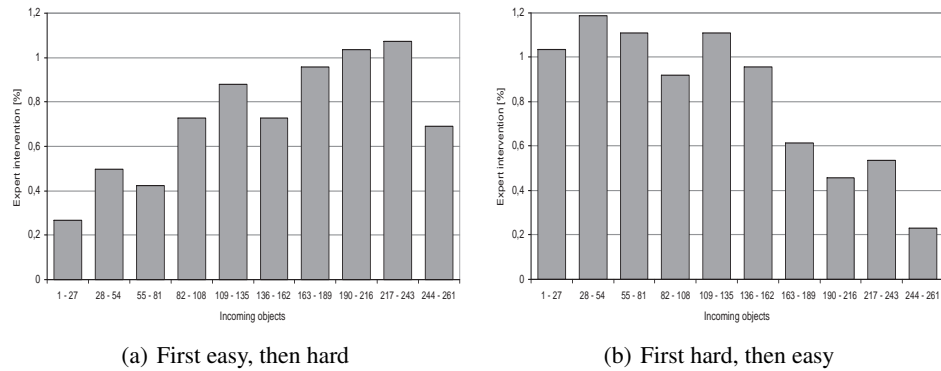


Fig. 3. Histogram of the percentage of objects that the expert has classified for spect data set, when the easiest (a) or hardest (b) objects arrived first. Objects are grouped by intervals of time.

4 Conclusions

A reinforcement learning-based approach has been devised for on-line learning and classification of incomplete objects in data streams. The proposed algorithm has some interesting properties. First, it is quite *general*: by automatically exploring a number of imputation/classification techniques which are available, the approach is able of outperforming each of these techniques being used isolatedly. Second, it is *memory effective*: by selectively choosing (guessing) the most representative objects, the size of the training set may be kept stable along time, in contrast to more naïve procedures which blindly include all incoming samples. Third, it makes a *modest usage of the expert knowledge*: by tapping into expert-provided class labels only on a few objects (less than 11%), the classification performance is significantly boosted.

Despite these good qualities, the algorithm has still some limitations that deserve further work. For instance, the exploratory component of the reinforcement learning implies trying many costly actions (such as imputation and classification) for each incoming object. Another interesting goal is designing and testing alternative formulations of the problem in terms of reinforcement learning.

References

1. Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data. Wiley (1987)
2. Ding, Y., Simonoff, J.S.: An investigation of missing data methods for classification trees applied to binary response data. *J. of Machine Learning Res.* **11** (2010) 131–170
3. Farhangfar, A., Kurgan, L., Dy, J.: Impact of imputation of missing values on classification error for discrete data. *Pattern Recognitions* **41**(12) (2008) 3692–3705
4. Millán-Giraldo, M., Sánchez, J.S., Traver, V.J.: Exploring early classification strategies of streaming data with delayed attributes. In: *Neural Information Processing*. Volume 5863 of *Lecture Notes in Computer Science.*, Springer (2009) 875–883
5. Vogiatzis, D., Stafylopatis, A.: Reinforcement learning for rule extraction from a labeled dataset. *Cognitive Systems Research* **3**(2) (2002) 237–253
6. Langford, J., Zadrozny, B.: Relating reinforcement learning performance to classification performance. In: *In Proc. of the Intl. Conference on Machine Learning*. (2005) 473–480
7. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. MIT Press (1998)
8. Chapelle, O., Schölkopf, B., Zien, A., eds.: *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. The MIT Press (2006)
9. Bruzzone, L., Roli, F., Serpico, S.B.: An extension of the Jeffreys Matusita distance to multiclass cases for feature selection. *IEEE Transactions on Geoscience and Remote Sensing* **33**(6) (1995) 1318–1321
10. Nagy, G.: Classifiers that improve with use. In: *In Proc. Conf. on Pattern Recognition and Multimedia*. (2004) 79–86
11. Frank, A., Asuncion, A.: UCI machine learning repository
12. Ripley, B.D.: *Pattern Recognition and Neural Networks*. Cambridge University Press (1996)
13. Library: Real medical data sets http://www.bangor.ac.uk/~mas00a/activities/real_data.htm.