Learning and Forgetting with Local Information of New Objects

Fernando D. Vázquez[1], J. Salvador Sánchez[2], Filiberto Pla[2]

[1]Centro de Reconocimiento de Patrones y Minería de Datos, Universidad de Oriente,
Av. Patricio Lumumba s/n, Santiago de Cuba 90500 (Cuba)
fvazquez@csd.uo.edu.cu
[2]Dept. Llentguages i Sistemas Informàtics, Universitat Jaume I
12071 Castelló (Spain)
{pla, sanchez}@uji.es

Abstract. The performance of supervised learners depends on the presence of a relatively large labeled sample. This paper proposes an automatic ongoing learning system, which is able to incorporate new knowledge from the experience obtained when classifying new objects and correspondingly, to improve the efficiency of the system. We employ a stochastic rule for classifying and editing, along with a condensing algorithm based on local density to forget superfluous data (and control the sample size). The effectiveness of the algorithm is experimentally evaluated using a number of data sets taken from the UCI Machine Learning Database Repository.

Keywords: Learning; Classification; Forgetting; Editing; Condensing

1 Introduction

In the context of pattern recognition, learning algorithms have been traditionally sorted into two broad categories: supervised and unsupervised, depending on whether labeled data are available or not. In a supervised scenario, the learner is mainly based on the information supplied by a set of labeled instances (training set, TS) that are assumed to correctly represent all the relevant classes. Violation of this assumption may seriously deteriorate the final classification accuracy achieved by the learning system.

Supervised classification methods usually operate in two steps: a) the *learning or training phase*, for the system to acquire the necessary knowledge from the labeled instances to make itself able to differentiate among the regarded classes; and b) the *classification or operational phase*, wherein the system proceeds to identify new unknown cases as members of the considered classes. Second stage is not started before completion of the first one and thereafter, no new knowledge is attained.

In the unsupervised learning problem, the learner is provided with only unlabelled examples. The task is to find "clusters" or groups of similar cases that probably correspond to the underlying classes. Unsupervised learning is often applied to discover structure, regularities or categories in the data, but typically requires human

analysis to determine whether the discovered regularities are interesting, and to determine the true correspondence between clusters and meaningful categories.

Since the early 90's a third approach to learning, namely *semi-supervised* (or *partially supervised*), has received much attention [1-4]. This paradigm conceptually represents a compromise between supervised and unsupervised learning, thus using a (generally) small number of labeled instances together with a (possibly) large set of unlabelled samples. Relevance of partially supervised learning systems is due to the fact that in many practical applications, collecting labeled training instances can be costly and time-consuming, while it is frequently easy to obtain unlabelled examples. Consequently, it results interesting to develop algorithms capable of employing both labeled and unlabelled data for classification.

This paper presents an idea to implement a classification system that not only can learn by operating with the labeled training instances, but could also benefit from the experience obtained when classifying new unlabelled patterns. The approach for working with an *ongoing learning* capability presents some interesting advantages: the classifier is more robust because errors or omissions in the original TS can be further corrected during operation and on the other hand, the system is capable to continue adapting itself to a possibly changing (non-stationary) environment.

The ultimate aim is to facilitate the learning system to progressively increase its knowledge and consequently, to enhance the final classification accuracy. In our proposal, a new classification rule based on class probabilities is employed as the central classifier. Because a basic goal is to make the ongoing learning procedure as automatic as possible, it has been designed to work by incorporating new examples into the TS after they have been labeled by the own system. This way, however, presents the danger of performance deterioration by the inclusion of potentially mislabeled patterns to the TS. In order to minimize the risk of introducing these errors, we will employ a stochastic editing algorithm that detects and discards mislabeled cases. Finally, in order to control de TS size we employ a new condensing technique based on local density.

Dasarathy [5] proposed a system with the ability of adapting to changing environments by employing the nearest neighbor (NN) rule as the central classifier and techniques to avoid the indiscriminate growth of the TS, or to prevent the degradation of its performance. Nevertheless, the main difference with respect to our proposal refers to the fact that Dasarathy's method involves the constant participation of a human expert to be in charge of the evaluation of the labels assigned by the system to new patterns and to decide which of them are to be incorporated to the training sample.

2 An Ongoing Learning Algorithm Using Class Probabilities

A basic goal of the learning system presented in this paper is to make it as automatic as possible. Accordingly, the procedure has been designed to work by incorporating new objects into the TS after they have been labeled by the own system (without the participation of a human expert). In order to use the information provided by the labeled samples, we employ a stochastic classification rule based on a neighborhood

criteria that takes into account both the distance of the neighbors to a sample and the probability of these neighbors to belong to each class [7].

However, it is evident that this working method can be self-defeating, in the sense that these new training elements will have the class label directly assigned by the decision rule. Therefore, there exists the risk to add several mislabeled cases on the TS and consequently, to degrade the overall system performance. The system we have designed attempts to overcome such a difficulty by employing a filter based on the stochastic classification rule mentioned in the previous paragraph.

On the other hand, albeit the original training instances are generally labeled by human experts (or, at least, under their supervision), it is still possible to introduce errors into the initial TS. Correspondingly, our first task will consist of looking for outliers (noisy, atypical and mislabeled patterns) in the TS in order to obtain a collection of correctly labeled examples.

Finally, by incorporating new objects into the TS, we may introduce redundant data, thus producing an increase in the computational cost of the system. In order to control the TS size, we employ a pruning or size reduction technique based on local density [6] to eliminate unnecessary training patterns.

In summary, the ongoing learning system will consist of the three main elements: a classifier to add new patterns into the TS, a filter or editing algorithm to clean the TS, and a pruning or condensing technique to control the TS size. The general procedure can be written as follows:

1. A first filter is applied to the original TS in order to remove possible noisy instances. The resulting edited set will be here referred to as *base knowledge*.
2. Classification of new objects (individually or in batches) starts with the base knowledge working as the current TS.
3. The set of new labeled patterns (those classified during the previous step) is now edited in order to detect possible misclassifications. The patterns identified as erroneous by the filtering algorithm will be removed from that set.
4. The base knowledge is now updated by incorporating the new labeled patterns that have not been discarded in the previous step. The resulting set is referred as to *current knowledge*.
5. The current knowledge is now edited in order to detect erroneous decisions made in Step 4
6. If the size of CA is greater than a certain size N, then employ a condensing algorithm.
7. Take CA as the current knowledge. If there are samples (or batches of samples) to classify, go to step 2, else stop.

An alternative to this scheme can be as follows: after editing in Step 3, we apply a condensing algorithm so that we add just few representative samples into the base knowledge and consequently, in this case Step 6 will not be necessary. In the experiments, the general algorithm will be referred to as V1, whereas the second alternative will be named V2.

Note that the original base knowledge (i.e., the initial TS) constitutes the only supervised element of our ongoing learning system. The unsupervised component comes from the unlabelled patterns that are sequentially taken (classified) by the own system.

2.1 A Stochastic Classification Rule

The rationale of this approach is aimed at using a classification rule based on local information of an instance like the *k*-NN, but considering the form of the underlying probability distribution in the neighborhood of a point. In order to estimate the values of the underlying distributions, we can use the distance between the sample and the prototypes. Given a sample, the closer a prototype, the more likely this sample belongs to the same class as the one of such a prototype.

Therefore, let us define the probability $P_i(\mathbf{x})$ that a sample $\mathbf{x}$ belongs to a class w_i as:

$$P_i(\mathrm{x}) = \sum_{j=1}^{k} p_i^j \frac{1}{(\varepsilon + d(\mathrm{x}, x^j))} \tag{1}$$

where p_i^j denotes the probability that the *k*-nearest neighbor x^j belongs to class w_i. Initially, the values of p_i^j for each prototype are set to 1 for its class label assigned in the TS, and 0 otherwise. These values could change in case an iterative process is used, but this is not the case in the approach we are presenting here.

The meaning of the above expression states that the probability that a sample $\mathbf{x}$ belongs to a class w_i is the weighted average of the probabilities that its *k*-nearest neighbors belong to that class. The weight is inversely proportional to the distance from the sample to the corresponding *k*-nearest neighbor. After normalizing,

$$p_i(\mathrm{x}) = P_i(\mathrm{x}) / \sum_{j=1}^{M} P_j(\mathrm{x}) \tag{2}$$

the class w_i assigned to a sample $\mathbf{x}$ is estimated by the decision rule

$$\delta_{\text{k-prob}}(\mathrm{x}) = w_i \ ; \quad w_i \ / \ p_i(\mathrm{x}) = \arg\max_j (p_j(\mathrm{x})) \tag{3}$$

The meaning of this expression is not more than the sample $\mathbf{x}$ will be assigned to the class with the highest probability, taking into account the contribution of the probabilities of belonging to each class of their neighbors, and the distances from the nearest neighbors to the sample $\mathbf{x}$.

2.2 A Filter to Clean the Current Knowledge

Following the general scheme of Wilson's editing [8], the technique employed in our ongoing learning algorithm consists of eliminating from the TS those instances whose label does not coincide with that assigned by the corresponding decision rule. In this case, the classifier used is that based on class conditional probabilities presented in Section 2.1.

2.3 A Forgetting Mechanism to Control the Size

In order to avoid the rapid growth of the TS due to the incorporation of new objects, we need to pick up a small number of representative samples. To this end, we search for regions of high density within the TS and then, the objects with maximal local density inside each region will be selected to constitute the condensed set.

The local density of each training pattern can be computed employing the expression

$$p(\mathbf{x}) = \sum_{\mathbf{x}_i \in C_j} \frac{1}{\varepsilon + d(\mathbf{x}, \mathbf{x}_i)} \tag{4}$$

where C_j is the label of $\mathbf{x}$ in the TS.

Using this, the condensing algorithm can be written as follows [6]:

1. Assign each point to an unitary set
2. For each class C_j in the TS do:
   - 2.1 For each $x \in C_j$ do
     - 2.1.1. Calculate its k nearest neighbors inside C_j
     - 2.1.2. Calculate the value $p(x)$ according to the previous formula
   - 2.2 For each $x \in C_j$ do:
     - 2.2.1. Find x_j so that $p(x_j) = \max p(x_i)$, $i = 1, \ldots, k$
     - 2.2.2. If $p(x_j) \geq p(x)$, join the class of x with the class of x_j
   - 2.3 Select in each group obtained in the previous step the point with maximal local density
3. Construct the condensed set with the points obtained in Step 2.3 with their original labels

3 Experimental Results

In our experiments, we have used six different databases taken from the UCI Machine Learning Database Repository (`http://www.ics.uci.edu/~mlearn`). To simulate the ongoing learning process, each of these databases was randomly divided into a number of blocks, each keeping the corresponding a priori class distribution. One of these batches of the partition was taken at random as the initial TS and another

as a test set to evaluate the effectiveness of the learning system; the remaining blocks were used to simulate the flow of untagged objects that arrive at the classifier.

To test the performance of the two variants described in Section 2 we have utilized the NN decision rule, taking the result of applying the algorithm to each block of new objects as the current TS. In the figures, we illustrate the percentage of correct classifications at each iteration (block of new objects).

As a complementary measure of the effectiveness of our algorithms, we have added the so-called "Learning Curve", which has been obtained as follows: First, both the initial TS and each of the batches (taken in the same order as they appear in the learning process) are edited and joined to the previous set; after each union, we compute the percentage of correct classifications using the NN rule.

In Fig. 1, we have the results for Australian and Cancer databases. In the case of Australian, results of both variants (V1 and V2) are similar, that is, the curve is growing up as the number of batches are processed. Thus, although the performance of V2 is lower than that of V1, both methods are able to improve the learning system quality by incorporating new knowledge. For Cancer database, there is a small difference with respect to the previous results: the curve of V2 is not always growing up. Despite this, it is important to note that in all cases, the performance of V1 and V2 is higher than that of the learning curve.

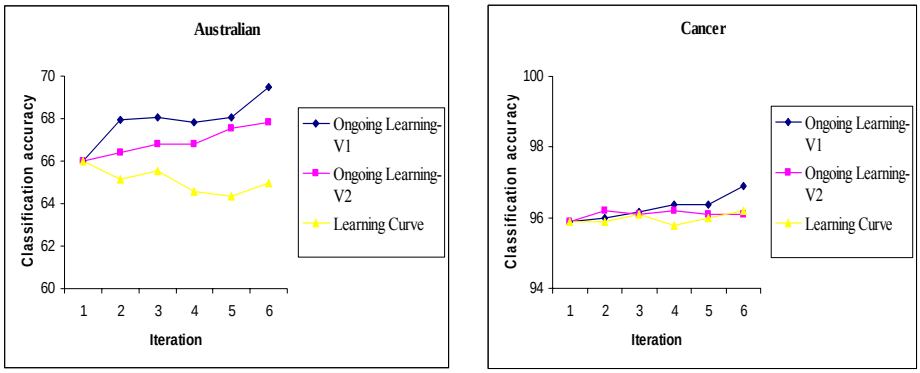

Fig. 1. Results for Australian (left) and Cancer (right) databases.

Figure 2 shows the results for Diabetes and Heart databases. In these two domains, the results are V1 are clearly better than those of V2. In some cases, the performance of V2 is even lower than that of the learning curve. This observation suggests that for these databases, it would be preferable to incorporate all the edited objects into the current TS, instead of pruning the result. For these particular databases, this can be due to the small size of the initial TS and the few number of new objects processed at each iteration. In fact, it is well-known that the use of some condensing technique over small data sets generally produces a significant degradation of the classifier performance. Therefore, one should take care of these situations in order to decide when to use or not a reduction technique.

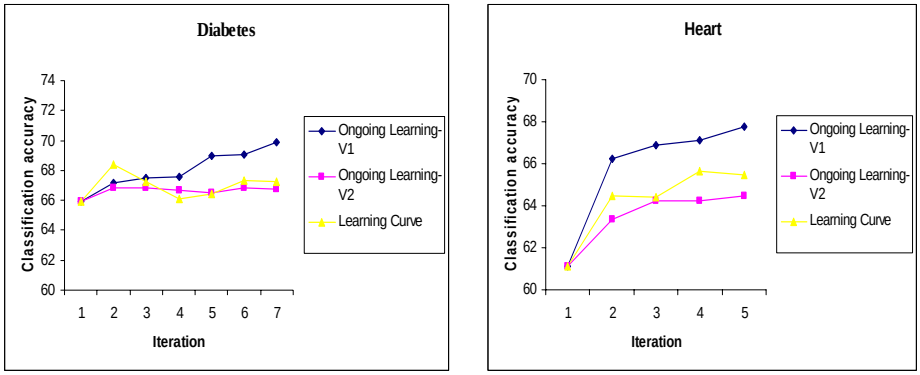

Fig. 2. Results for Diabetes (left) and Heart (right) databases.

In Figure 3, we have plotted the results for Phoneme and Texture databases. It has to be mentioned that these are moderate sized databases. In both these domains, incorporation of new objects into the TS produces an increase in performance with respect to that of the initial knowledge, although such an improvement is not significant. On the other hand, at each iteration the result of the learning curve is better than the rates obtained by V1 and V2 algorithms. When comparing both variants, the results suggest that V2 is somewhat better than V1, especially in the case of the Texture database.

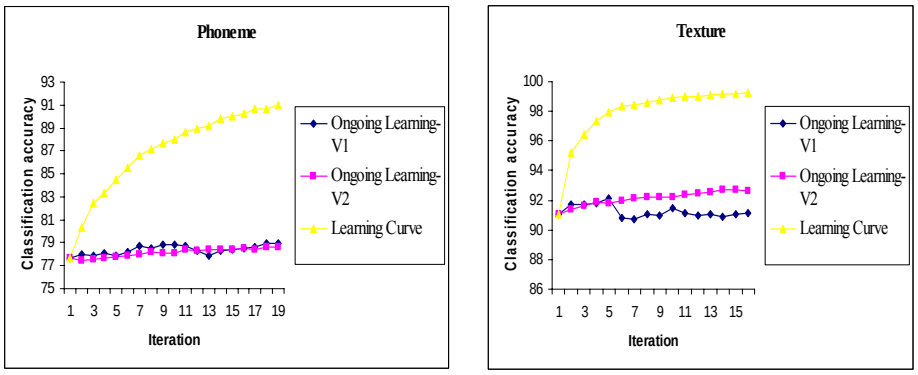

Fig. 3. Results for Phoneme (left) and Texture (right) databases.

As a summary of the results given in this section, it can be said that the use of a filter technique and a condensing algorithm within the ongoing learning system allows to increase the quality of the initial TS. The V2 alternative seems to be superior for those databases with a higher number of initial training samples, whereas V1 works better for small data sets.

4 Conclusions and Future Work

In this paper, an ongoing learning algorithm to increase the performance of the knowledge in partially supervised environments has been introduced. It makes use of a reduced number of labeled instances and a possibly large amount of unlabelled objects. The system includes some tools that allow us to filter the new knowledge acquired during operation, thus avoiding the risk of incorporating several mislabeled patterns into the TS and consequently, to degrade the overall system accuracy. Also, a pruning technique is used in the system in order to control the TS size by removing redundant patterns. In the empirical evaluation, the results have shown that in general the objects incorporated to the knowledge are able to improve the system performance given by the original TS.

As future work, we can suggest to develop schemes similar to those proposed in this work, but for mixed data (categorical and numerical). In this context, ensembles of classifiers could be especially taken into account as a way of handling data with mixed attributes.

Acknowledgments. This work has been partially supported by projects DPI2006–15542 from the Spanish CICYT and CSD2007-00018 from Consolider-Ingenio 2010 (Ministerio de Educación y Ciencia).

References

1. Barandela, R., Juárez, M.: Ongoing learning for supervised pattern recognition. In: 14th Brazilian Symposium Computer Graphics and Image Processing, pp. 51-58 (2001)
2. Bensaid, A.M., Hall, L.O., Bezdek, J.C., Clarke, L.P.: Partially supervised clustering for image segmentation. Pattern Recognition 29, 859-871 (1996)
3. Blum, A., Chawla.: Learning from labelled and unlabeled data using graph mincuts. In: 18th International Conference on Machine Learning, pp. 19-26 (2001)
4. Castelli, V., Cover, T.M.: On the exponential value of labeled samples. Pattern Recognition Letters, 16, 105-111 (1995)
5. Dasarathy, B.V.: Adaptive decision systems with extended learning for deployment in partially exposed environments. Optical Engineering 34, 1269-1280 (1995)
6. Pascual, D.; Pla, F., Sánchez, J.S.: Non Parametric Local Density-based Clustering for Multimodal Overlapping Distributions. In: Intelligent Data Engineering and Automated Learning, LNCS, vol. 4224, pp. 671-678. Springer, Heidelberg (2006)
7. Vázquez F, Sánchez J.S., Pla F.: A stochastic approach to Wilson's editing algorithm. Pattern Recognition and Image Analysis. In: Pattern Recognition and Image Analysis, LNCS, vol. 3523, pp. 35-42. Springer, Heidelberg (2005)
8. Wilson, D.L.: Asymptotic properties of nearest neighbour rules using edited data. IEEE Trans. on Systems, Man and Cybernetics 2, 408-421 (1972)