Nearest neighbor learning by means of labelled and unlabelled data

F. Vázquez†, J.S. Sánchez‡, F. Pla‡
†Dept. de Computación, Universidad de Oriente
Av. Patricio Lumumba s/n, 90100 Santiago de Cuba, Cuba
E-mail: fvazquez@csd.uo.edu.cu
‡ Dept. de Llenguatges i Sistemes Informàtics, Universitat Jaume I
Av. Sos Baynat s/n, 12071 Castelló de la Plana, Spain
E-mail: {pla, sanchez}@uji.es

Abstract

A classification system with the capability of continuously increasing its knowledge during the operational phase is here discussed. This idea is strongly related to learning in partially supervised environments in the sense that at the start, the system has only a (possibly) reduced number of labelled instances, but this current knowledge will be progressively increased during the classification of new unlabelled patterns. The learning system proposed in the present paper is based on the popular nearest neighbor classifier and some related techniques. The effectiveness of the algorithm is experimentally evaluated using some benchmark data sets taken from the UCI Machine Learning Database Repository.

1 Introduction

Learning algorithms have been traditionally sorted into two broad categories: supervised and unsupervised, depending on whether labelled data is available or not. In a supervised scenario, the learner is based on the information supplied by a a set of labelled 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.

Supervised classification methods usually operate in two steps: a) the *learning or training phase*, for the system to acquire the necessary knowledge from the labelled 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 correspondence between clusters and meaningful categories.

Since the early 90's a third approach to learning, namely *partially supervised*, has received much attention [2–4, 14, 15, 18]. This paradigm conceptually represents a compromise between supervised and unsupervised learning, thus using a (generally) small number of labelled 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 labelled 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 labelled and unlabelled data for classification. Learning from partially labelled data is also referred to as *semi-supervised learning* [1, 13].

This paper presents an idea to implement a classification system that not only can learn by operating with the labelled training instances, but could also benefit from the experience obtained when classifying new unlabelled patterns. The approach for working with "ongoing learning" presents some advantages: the classifier is more robust because errors or omissions in the original TS can be further corrected during operation, and the system is capable to continue adapting itself to a possibly changing 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, the nearest neighbor (NN) rule is employed as the central classifier, mainly because of its flexibility. 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 labelled by the own system. This way, however, presents the danger of performance deterioration by the inclusion of potentially mislabelled patterns to the TS. In order to minimize the risk of introducing these errors, we will employ some filters that detect and discard those mislabelled cases.

From now on, the rest of the paper is organized as follows. Section 2 provides a general description of the k-NN rule along with the most important pros and cons of using this classifier. Section 2 also describes an editing algorithm based on an estimation of probabilities. In Sect. 3, we introduce the ongoing learning system proposed in the present paper. Next, Sect. 4 provides the results obtained from a preliminary empirical study. Finally, the main conclusions and possible directions for future research are outlined in Sect. 5.

2 The k-nearest neighbors classifier

One of the most widely studied supervised classification approaches corresponds to the k-NN decision rule [6]. In brief, given a set of n previously labelled examples, say $X = \{(x_1, \omega_1), (x_2, \omega_2), \ldots, (x_n, \omega_n)\}$, the k-NN classifier consists of assigning an input sample x to the class most frequently represented among the k closest instances in the TS, according to a certain similarity measure (generally, the Euclidean distance metric). A particular case of this rule is when $k = 1$, in which an input sample is decided to belong to the class indicated by its closest neighbor.

Several properties make the k-NN classifier quite attractive, including the fact that the asymptotic risk (i.e., when $n \rightarrow \infty$) tends to the optimal Bayes risk as $k \rightarrow \infty$ and $k/n \rightarrow 0$ [5]. If $k = 1$, the upper bound of the classification error rate is approximately twice the Bayes error [6]. The optimal behavior of this rule in asymptotic classification performance along with a conceptual and implementational simplicity make it a powerful classification technique capable of dealing with arbitrarily complex problems, provided that there is a large enough number of training instances available.

However, in many practical situations, such a theoretical maximum can hardly be achieved due to certain inherent weaknesses that significantly reduce the effective applicability of k-NN classifiers. In particular, the performance of these rules, as with any non-parametric classification approach, is extremely sensitive to data complexity [7].

For example, classification accuracy of k-NN classifiers significantly drops down in domains where many data attributes are irrelevant [16]. Such attributes inappropriately affect the values returned by most dissimilarity metrics. Another problem using the k-NN rule refers to the seeming necessity of a lot of memory and computational resources (especially, in applications with a huge number of training examples). Moreover, these classifiers cannot be straightforwardly employed in domains with missing attributes. Also, the class imbalance (i.e., high differences in class distributions) has been reported as an obstacle on applying distance-based algorithms to real-world problems [11].

On the other hand, class overlapping and noise or imperfections in the TS negatively affect the performance of the k-NN classifiers, and this has been widely demonstrated in many empirical studies (e.g., see [17]). That is the reason why a considerable amount of works have been devoted to improve the classification accuracy by eliminating outliers from the original TS and also cleaning possible overlapping between classes. This strategy has generally been referred to as *editing* [9].

The general idea behind almost any editing procedure consists of estimating the true classification of instances in the TS to retain only those which are correctly labelled. Differences among most editing schemes refer to the classification rule employed for editing purposes along with the error estimate and the stopping criterion [10].

The first proposal to select a representative subset of labelled instances corresponds to Wilson's editing [21], in which a k-NN classifier is used to keep in the TS only ”good” examples (that is, training instances that result correctly classified by the k-NN rule). Tomek [19] extended this scheme with a procedure that utilized all the l-NN classifiers, with l ranging from 1 through k, for a given value of k.

A slight modification of the original Wilson's algorithm consists of using, instead of the k-NN classifier, an alternative rule based on the k nearest centroid neighbors (k-NCN) [17], which has been proven to be superior to the traditional k-NN classifier in many practical situations. This kind of neighborhood is defined taking into account not only the proximity of instances to a given input pattern but also their symmetrical distribution around it.

2.1 Estimating class conditional probabilities for editing

Recently, new editing schemes have been proposed, in which the elimination rule is based on an estimation of the probability of each training instance to belong to a certain class, that is, considering the form of the underlying probability distribution in the neighborhood of a point [20]. In order to estimate the values of these distributions, we can compute the distance between a given sample and the training instances.

Given a sample, the closer an instance, the more likely this sample belongs to the same class as the one of such an instance. Accordingly, let us define the probability $P_i(x)$ that a sample x belongs to a class i as:

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

where p_i^j denotes the probability that the k nearest neighbor x^j belongs to class i, and δ represents a certain distance function. Initially, the values of p_i^j for each instance are set to 1 for its class label assigned in the TS, and 0 otherwise.

The meaning of the above expression states that the probability that a sample x belongs to a class 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 neighbors. From this, we can derive a new decision rule, namely k-Prob, in which a new sample x will be assigned to the class whose probability $P_i(x)$ is maximum.

Following the general scheme of Wilson's editing, the new algorithms consist of eliminating from the TS those instances whose label does not coincide with that assigned by the decision rule based on class conditional probabilities (k-Prob).

A further extension to this proposal consists of considering a threshold, $0 < \mu < 1$, in the classification rule, with the aim of eliminating those instances whose probability to belong to the class assigned by the rule is not significant. Correspondingly, we are removing samples from the TS that are in the decision borders, where the class conditional probabilities overlap and are confusing, in order to obtain edited sets whose instances have a high probability of belonging to the class assigned in the TS.

3 The use of unlabelled data to increase knowledge

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 patterns into the TS after they have been labelled by the own system (without the participation of a human expert). However, it is evident that this working method can be self-defeating, in the sense that these new training elements would have the class label directly assigned by the decision rule. Therefore, there is the risk to incorporate several mislabelled cases into the TS and consequently, to degrade the overall system accuracy. The system we have designed attempts to overcome such a difficulty by employing some editing algorithms.

On the other hand, albeit the training instances are generally labelled by human experts (or, at least, under their supervision), it is possible to introduce errors into the TS. Thus our initial task will consists of looking for outliers in the TS in order to obtain a collection of correctly labelled instances. In summary, the learning procedure for partially supervised domains consists of the following steps:

1) Initial TS is stored in memory.

2) A first filter is applied to the original TS in order to remove possible noisy instances. As a by-product, it also produces a reduction in the TS size. The resulting edited set will be here referred to as *base knowledge*.

3) Classification phase starts with the base knowledge as the TS.

4) The set of new labelled patterns (those classified during the previous step) is now edited in order to detect possible misclassifications. The patterns identified as erroneous by the editing algorithm will be removed from that set.

5) The base knowledge is now updated by incorporating the new labelled patterns that have not been discarded in the previous step.

6) Return to Step 3 with the new base knowledge.

For the filters considered in this procedure, one could employ any editing algorithm. In the present paper, we have applied two of the schemes introduced in Sect. 2: the k-NCN editing, and the first algorithm based on class conditional probabilities, namely Wilson-Prob [20]. Analogously, the classification phase (Step 3) can be performed by applying any classifier. Here we have used the classical k-NN rule, the k-NCN classifier, and the new k-Prob decision scheme.

Note that the original base knowledge constitutes the only supervised element of our learning system. The unsupervised component comes from the unlabelled patterns that are sequentially classified and edited by the own system.

Dasarathy [8] proposed a decision system with a design very related to ours. He was also concerned with the robustness of the system through varying domains and with the problem of unrepresentative pre-training. The latter is what he called ”partially exposed environments”. Consequently, Dasarathy presented an on-line adaptive learning system with two capabilities: a) to progressively improve the classification of patterns belonging to the known classes and, b) to detect the objects not belonging to the currently known classes

However, Dasarathy’s system requires the steady 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 into the TS. Unfortunately, in real-world operational phase, such operator supervision may be unavailable. We avoid this bottleneck by including in our procedure the necessary tools to allow the system to decide which pieces of new knowledge are trustworthy enough to be accepted.

4 Experimental results

In our experiments, we have included four data sets taken from the UCI Machine Learning Database Repository (`http://www.ics.uci.edu/~mlearn`). A number of different partitions were randomly produced for each data set, all keeping the a priori class probabilities. One of these partitions is used as the initial TS, one as an independent validation set, and the rest will be employed as sets of unlabelled data in order to simulate the sequence required for developing the capacity of increasing the knowledge by means of the algorithm presented in the previous section.

Data set	Classes	Features	Size	% Class 1	% Class 2	Partitions
Breast	2	9	683	65.2	34.8	10
Diabetes	2	8	786	34.9	65.1	11
German	2	24	1002	70.4	29.6	14
Heart	2	13	270	55.6	44.4	9

Table 1: A brief summary of the UCI databases used in the experiments.

The main characteristics of the data sets used in the present experiments are summarized in Table 1. The column ”Partitions” indicates the total number of random partitions produced for each database. This number means that, for example, in Breast database the classification system will have 8 opportunities to increase its base knowledge, that is, the number of sets with unlabelled data. By this, it is evident that the amount of labelled instances is much smaller than that of the unlabelled patterns. The reason is that, as already stated in Sect. 1, in real applications collecting labelled examples often becomes a costly and difficult process, thus we are here reproducing this practical situation.

The experiments consist of comparing the 1-NN classification accuracy when using the initial TS with that obtained when incorporating the new labelled patterns to the TS after processing each of the partitions. The aim is to evaluate the capacity of increasing the knowledge with the application of our learning algorithm in a partially supervised environment.

t	Alg1	Alg2	Alg3	Alg4
0	92.54	92.54	92.54	92.54
1	94.03	94.03	**94.03**	94.03
2	94.03	94.03	94.03	94.03
3	94.03	94.03	94.03	94.03
4	**95.52**	94.03	92.54	**95.52**
5	95.52	94.03	92.54	95.52
6	95.52	94.03	92.54	95.52
7	95.52	**95.52**	94.03	95.52
8	95.52	95.52	94.03	95.52
1-NN	92.48			

Table 2: Classification accuracies for Breast database (1-NN indicates the classification accuracy when using the original TS without any editing).

Tables 2, 3, 4 and 5 provide the classification accuracies over the different databases used in the present experiments. Column t refers to each partition included in the process. Thus $t = 0$ represents the initial base knowledge, that is, the original TS after being edited. The set obtained at any time $t > 0$ is then incorporated into the previous knowledge (the set of instances available at time $t - 1$). For example, in $t = 1$ the current knowledge refers to that acquired in $t = 0$, and it is now employed to classify the first set of unlabelled patterns. After classifying, we edit the new labelled instances in order to discard possible misclassifications. Then the current knowledge is updated by including the instances that have not been eliminated in editing. The result will be the input set in $t = 2$.

The meaning of Alg1, Alg2, Alg3, and Alg4 in Tables 2, 3, 4 and 5 is as follows. In the case of Alg1, we have employed the k-NCN algorithm for editing and the k-NN rule for classification. Alg2 uses the k-NCN algorithm both for editing and for classifying new patterns. Alg3 applies Wilson-Prob for editing and the k-Prob decision rule for classification. Finally, Alg4 is equal to Alg3, but using the nearest centroid neighborhood instead of the classical nearest neighborhood. Values in bold type indicate the first occurrence of the highest accuracy for each algorithm and each database.

t	Alg1	Alg2	Alg3	Alg4
0	66.67	66.67	**70.24**	68.45
1	66.07	66.07	70.24	69.05
2	66.07	68.45	69.64	69.64
3	66.07	69.64	67.86	69.64
4	65.48	69.05	67.26	69.64
5	65.48	69.05	67.86	69.64
6	66.07	69.64	67.86	69.64
7	66.67	70.24	67.86	70.24
8	**67.26**	**70.83**	67.86	**70.83**
9	67.26	68.45	66.67	70.24
1-NN	66.32			

Table 3: Classification accuracies for Diabetes database (1-NN indicates the classification accuracy when using the original TS without any editing).

From the results reported in Tables 2 and 3, some conclusions can be drawn. First, it has to be noted that all implementations outperform the 1-NN classification accuracy reported as a baseline. On the other hand, except Alg3 when applied over Diabetes and German databases, all the other cases show a certain improvement in performance with respect to the original edited TS ($t = 0$). Nonetheless, in terms of accuracy, it seems difficult to decide which learning algorithm is the best. In practice, any of those three algorithms (Alg1, Alg2, and Alg4) could constitute a good solution for increasing the knowledge in a partially supervised environment.

It is worth pointing out the fact that in general, the system obtains a maximum value in performance after processing a relatively small number of partitions. This is important because it can mean that after a number of iterations, the inclusion of more instances does not provide more information to the system. In this situation, the system increases the size of the TS, but without increasing its knowledge. This is a crucial issue that will be investigated in further extensions to this work.

t	Alg1	Alg2	Alg3	Alg4
0	67.61	67.61	**71.83**	69.01
1	69.01	69.01	71.83	69.01
2	**70.42**	**70.42**	71.83	69.01
3	70.42	70.42	71.83	69.01
4	70.42	70.42	70.42	69.01
5	70.42	70.42	67.61	**70.42**
6	67.61	70.42	67.61	70.42
7	67.61	70.42	69.01	70.42
8	67.61	70.42	69.01	70.42
9	67.61	70.42	69.01	70.42
10	67.61	70.42	70.42	70.42
11	67.61	70.42	70.42	70.42
12	67.61	70.42	70.42	70.42
1-NN	65.81			

Table 4: Classification accuracies for German database (1-NN refers to the classification accuracy when using the original TS without any editing).

t	Alg1	Alg2	Alg3	Alg4
0	51.61	51.61	54.84	54.84
1	61.29	58.06	58.06	61.29
2	61.29	**64.52**	**64.52**	61.29
3	64.52	64.52	64.52	64.52
4	**67.74**	64.52	64.52	64.52
5	67.74	64.52	64.52	64.52
6	67.74	64.52	64.52	64.52
7	67.74	64.52	64.52	**67.74**
1-NN	53.33			

Table 5: Classification accuracies for Heart database (1-NN refers to the classification accuracy when using the original TS without any editing).

5 Conclusions and further extensions

In this paper, a learning algorithm to increase the knowledge in partially supervised environments has been introduced. It makes use of a reduced number of labelled instances and a (possibly) large amount of unlabelled patterns. The system includes a set of tools allowing to filter the new knowledge acquired during operation. Thus we pursue to avoid the risk of incorporating several mislabelled patterns into the TS and consequently, to degrade the overall system performance.

In the empirical evaluation of the learning system, we have used different classification rules and several editing algorithms. Except in the case of employing a scheme based on class conditional probabilities for both classification and editing (Alg3), all the other alternatives have been proven to perform well enough for increasing the knowledge.

An important issue related to the performance of a system with the capability of increasing its knowledge refers to the possibility for the TS size to grow too much and consequently, some problems related to storage space and classification time can make such a system useless. Although editing has the property, as a by-product, of reducing the TS size, this is not achieved in a considerable amount. Accordingly, possible extensions to this work are in the direction of including some techniques to intelligently reduce the TS size. To this end, both adaptive and selective condensing algorithms [12] can be of interest to control the TS size.

Also, the possibility of discovering new classes not present in the original TS can result important for this kind of learning systems in partially supervised domains. Therefore, future research includes the study of unsupervised techniques in order to incorporate this additional capability into our learning system.

Acknowledgments

This work has been partially supported by grant TIC2003-08496 from the Spanish CICYT (Ministerio de Ciencia y Tecnología).

References

[1] Belkin, M., Niyogi, P.: Semi-supervised learning on Riemannian manifolds, Machine Learning **56** (2004) 209–239.

[2] Bensaid, A.M., Hall, L.O., Bezdek, J.C., Clarke, L.P.: Partially supervised clustering for image segmentation, Pattern Recognition **29** (1996) 859–871.

[3] Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts, In: Proc. 18th. Intl. Conf. on Machine Learning (2001) 19–26.

[4] Castelli, V., Cover, T.M.: On the exponential value of labeled samples, Pattern Recognition Letters **16** (1995) 105–111.

[5] Cover, T.M.: Estimation by the nearest neighbor rule, IEEE Trans. on Information Theory **14** (1968) 50–55.

[6] Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification, IEEE Trans. on Information Theory **13** (1967) 21–27.

[7] Dasarathy, B.V.: Nearest Neighbor Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamos, CA (1991).

[8] Dasarathy, B.V.: Adaptive decision systems with extended learning for deployment in partially exposed environments, Optical Engineering **34** (1995) 1269–1280.

[9] Devijver, P.A., Kittler, J.: Pattern Recognition: A Statistical Approach. Prentice Hall, Englewood Cliffs, NJ (1982).

[10] Ferri, F.J., Albert, J.V., Vidal, E.: Considerations about sample-size sensitivity of a family of edited nearest-neighbor rules, IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics **29** (1999) 667–672.

[11] Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study, Intelligent Data Analysis **6** (2002) 429–449.

[12] Kim, S.-W., Oommen, B.J.: A brief taxonomy and ranking of creative prototype reduction schemes, Pattern Analysis & Applications **6** (2003) 232–244.

[13] Krogel, M.A., Scheffer, T.: Multirelational learning, text mining, and semi-supervised learning for functional genomics. Machine Learning **57** (2004) 61–81.

[14] Mantero, P., Moser, G., Serpico, S.B.: Partially supervised classification of remote sensing images through SVM-based probability density estimation, IEEE Trans. on Geoscience and Remote Sensing **43** (2005) 559–570.

[15] Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM, Machine Learning **39** (2000) 103–134.

[16] Okamoto, S., Yugami, N.: Effects of domain characteristics on instance-based learning algorithms, Theoretical Computer Science **298** (2003) 207–233.

[17] Sánchez, J.S., Barandela, R., Marqués, A.I., Alejo, R., Badenas, J.: Analysis of new techniques to obtain quality training sets, Pattern Recognition Letters **24** (2003) 1015–1022.

[18] Szummer, M.O.: Learning from Partially Labeled Data, PhD thesis, Massachusetts Inst. of Technology (2002).

[19] Tomek, I.: An experiment with the edited nearest neighbor rule. IEEE Trans. on Systems, Man and Cybernetics **6** (1976) 448–452.

[20] Vazquez, F., Sánchez J.S., Pla, F.: A stochastic approach to Wilson's editing algorithm, In: Pattern Recognition and Image Analysis, Lecture Notes in Computer Science **3523** (2005) 35–42.

[21] Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data sets, IEEE Trans. on Systems, Man and Cybernetics **2** (1972) 408–421.