Ensembles of data-reduction-based classifiers for distributed learning from very large data sets

R. A. Mollineda, J. M. Sotoca, J. S. Sánchez

Dept. de Lenguajes y Sistemas Informáticos
Universidad Jaume I
Av. de Vicent Sos Baynat, s/n. 12071 Castellón. Spain
{mollined,sotoca,sanchez}@lsi.uji.es

Abstract. An ensemble of classifiers is a *set* of classification models and a *method* to combine their predictions into a joint decision. They were primarily devised to improve classification accuracies over individual classifiers. However, the growing need for learning from very large data sets has opened new application areas for this strategy. According to this approach, new ensembles of classifiers have been addressed to partition a large data set into possible disjoint moderate-sized (sub)sets, which are then distributed along with a number of individual classifiers across multiple processors for parallel operation. A second benefit of reducing sizes is to make feasible the use of well-known learning methods which are not appropriate for handling huge amount of data. This paper presents such a distributed model as a solution to the problem of learning on very large data sets. As individual classifiers, data-reduction-based techniques are proposed due to their abilities to reduce complexities and, in much cases, error rates.

Keywords: ensembles of classifiers, classifiers combination, parallel learning, data mining, very large data sets.

1 Introduction

Currently, many applications use extremely large data sets of high dimensionality, thus classifying, understanding or compressing this information becomes a very difficult task. Data and web mining, text categorization, financial forecasting, and biometrics are some examples of domains in which huge amounts of information have to be employed.

The processing of a very large data set suffers from a major difficulty: high storage and time requirements. On one hand, a large data set cannot be fully stored in the internal memory of a computer. On the other hand, the time needed to learn from such a whole data set can become prohibitive. These problems are specially critical in the case of using some distance-based learning algorithm, such as the Nearest Neighbor rule [1, 2], due to the apparent necessity of indiscriminately storing all the training instances.

Under these conditions, one solution consists of using some data reduction technique. This idea has been traditionally explored by many researchers, resulting in the development of many algorithms. Within this context, two general approaches can be remarked: *feature reduction* and *instance reduction*. Both aim at remove redundant and non-discriminant information from the original data set. While the former group selects/creates a minimal (sub)set of discriminant attributes, the latter selects/creates a minimal (sub)set of relevant instances for classification purposes. In both cases, the ultimate goal is to reduce the storage and time complexity, while maintaining the classification performance.

Most of the reduction algorithms reported in the literature have to process the data in its entirety to yield a small or medium-sized (sub)set. Paradoxically, in general it is not possible to apply these methods to huge databases because they cannot be entirely loaded in memory, or because their handling can become prohibitively expensive. Therefore, reduction techniques do not seem to be a straightforward solution to fully manage a huge data set.

An interesting alternative to process a large data set is to independently learn from a number of moderate-sized subsets and combine their results by means of an *ensemble of classifiers*. The idea is very intuitive and can be roughly summarized as follows: to create a number of (possible disjoint) subsets of a (very) large data sets, and distribute them along with independent classifiers across parallel processors for simultaneous learning.

Ensemble of classifiers constitutes one of the most promising current research directions in pattern recognition and machine learning. It basically consists of combining the results from multiple classifiers which potentially offered complementary information. The traditional goal of the ensemble has been to achieve higher classification accuracy than that produced by its components (the individual classifiers that make it up). This expectation is intuitively supported by the complementarity or diversity of these classifiers [3].

The focus of this paper is on using an ensemble of data reduction techniques to process several subsets of the original training set. Correspondingly, the new goal of the ensemble, which is not necessarily in conflict with the traditional one, is to distribute data and learning [4, 5]. These two papers introduce combined learning strategies that efficiently deal with disjoint subsets of large data sets, obtaining an overall predictive accuracy comparable to or better than that of a single classifier over the entire data set.

The rest of the paper is organized as follows. Section 2 introduces the main ensemble techniques based on resampling. Section 3 provides a brief review of some combining methods for feature and instance set reduction. Finally, Section 4 gives a summary of preliminary conclusions from theoretical and practical viewpoints, and a discussion on future research directions.

2 Ensembles methods by resampling

The disagreement among the predictions of single classifiers is a necessary condition for a successful combination. Therefore, methods for defining ensembles focus on producing diversity in classifiers operation. This effect can be achieved by altering training sets or by combining dissimilar classifiers (varying parameters or using different types of classifiers). This section describes two well-known methods for creating ensembles: Bagging [6] and Boosting [7]. They look for diversity by modifying training sets. Following this strategy, they randomly sample with replacement the entire data set (*resampling*) to obtain multiple training sets for a same number of classifiers. Each classifier is then trained using only the instances of its corresponding set.

Bagging [6] is an ensemble method which generates multiple independent training sets by randomly resampling the original data set. Each resulting training set may have many repeated training instances, while others may be left out. Each individual classifier is then trained with a different sampling of the data set. Because of the independence of training sets resampling, learning of individual classifiers can be accomplished simultaneously, for example, in different processors. Nevertheless, sizes of resulting training sets are generally closer to the size of the original data set (about 80-100%), being their sum greater than the total amount. Even assuming an ideal multiprocessing configuration, Bagging could yield a negligible (or zero) reduction of the total effort, which makes this technique not suitable for direct managing large data sets.

Boosting [7] represents a family of techniques. This method creates a *sequence* of training sets and classifiers. Each training set is selected based on the performances of previous classifiers. In each step of the sequence, the probability to choose a training instance is proportional to the number of times this instance has been incorrectly classified by previous classifiers. In that way, the new classifier is addressed to better learn those instances in which the current ensemble has poor performances. Regarding the sequential operation of Boosting, it is not possible a parallel computation, which makes this method also inappropriate for directly learning on large data sets.

Two representative methods of Boosting are Arcing [8] and AdaBoost [7, 9]. Both methods start from a uniform distribution of equal probabilities (*weights*) to select training instances.

Like Bagging, Arcing[1] resamples the data set to obtain a training set for each classifier $t + 1$. Unlike Bagging, no all training instances have the same probability to be chosen. Instances with higher misclassification rates, according to previous classifiers $1, 2, \ldots, t$ in the sequence, will have higher probabilities. Formally, given the instance x_i, $1 \leq i \leq N$, let n_i be the number of times that x_i has been misclassified by classifiers $1, 2, \ldots, t$. The probability p^i_{t+1} to select x_i for the classifier $t + 1$ in the next iteration is computed as:

$$p^i_{t+1} = \frac{1 + n_i^4}{\sum_{j=1}^{N}(1 + n_j^4)} \tag{1}$$

The exponent 4 was empirically selected after exhaustive experiments on the form $1 + n_i^h$, with $h \in \{1, 2, 4\}$ [8]. Higher values of h were not tested, so

[1] ARCing comes from *Addapted Resampling and Combining*. It was originally named as `arc-x4` by Breiman [8].

the technique does not exclude better results. In spite of its simplicity, Arcing produces accurate ensembles.

AdaBoost follows a different procedure. It may select training instances, or it may use all of them to train a classifier which makes use of weights in the learning process. The error ϵ_t of a single classifier t is computed as the sum of probabilities of the misclassified instances. An importance coefficient for the classifier t is then computed as:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right) \tag{2}$$

Note that $\alpha_t \geq 0$ if $\epsilon_t \leq \frac{1}{2}$ and, in general, α_t gets larger as ϵ_t gets smaller. The probabilities or weights for the next iteration are updated as follows,

$$p^i_{t+1} = \frac{p^i_t}{P_t} \cdot \begin{cases} e^{-\alpha_t} & \text{if } x_i \text{ is correctly classified by classifier } t \\ e^{\alpha_t} & \text{if } x_i \text{ is misclassified by classifier } t \end{cases} \tag{3}$$

where P_t is a normalization factor to sustain p_t as a distribution. The effect of rule (3) is to increase weights of misclassified instances and to decrease weights of correctly classified instances.

At recognition stage, AdaBoost classifies an unseen instance by a weighted majority vote considering importance coefficients α_t of all individual classifiers.

As commented, Bagging and Boosting are not adequate to manage large data sets. Some authors [4, 5] have proposed methods for an effective learning from disjoint subsets of a data set. In [5], a (large) data set is partitioned into disjoint subsets using both random and clustering methods. A preliminary experiment with an ensemble of classifiers on these partitions did not improve the accuracy of a single learner (C4.5 decision tree) using the entire data set. The next series of experiments involved the use of Bagging on each subset to improve its individual classification performance. Within this context, the final decision was defined in two steps: first, a majority decision was made for each subset (among its bags); second, the final vote was taken from subset decisions. Some conclusions of this work are: i) clustering partition is better than random and stratified partition, which are mostly equivalent; ii) combining learning on disjoint partitions do not improve (in general) the performance of a single learning on the entire data set; iii) bagging these subsets tends to improve the results of a single classifier learning on the entire data set. As a final summary of this paper can be stated that Bagging was able to regain the discrimination capacity lost by disjoint subsets with respect to the entire data set.

The purpose of this paper is to present the previous scheme (in its general form) as a solution for the problem of learning from a large data set. That is, i) to partition the entire data set into disjoint moderate-sized subsets; ii) to learn from each independent subset by a classification model which may perform in parallel; and iii) to combine the independent predictions into a joint decision. As individual classifiers, we would tend to favor data-reduction-based learning rules regarding both feature reduction and instance reduction.

3 Ensembles of data-reduction-based classifiers

Next two subsections present an overview about ensembles of classifiers based on data reduction. The first part is a review of combinations of learners from diverse feature subsets, while the second part is a discussion about ensembles that are based on instance reductions.

3.1 Feature set reduction

The study of the role of features in the analysis of huge data sets is an actual issue in multiple real world classification problems. Many data sets are made up of distributions of instances in high-dimensionality spaces. When such distributions are analyzed, it is usually observed redundant or noisy information in the values of features in terms of which each instance is described. Lots of methods have been reported to remove such irrelevant data, while maintaining or increasing the discriminant information of the resulting data set.

In the context of ensembles of classifiers, feature reduction methods have been also used to generate diversity in training data subsets, which can be created from qualitative different subsets of features. According to [10], feature partitioning can produce more complementarity than instance partitioning. The most common example of feature partitioning are randomized attributes subsets, which is a simple method to increase the independence among classifiers [11, 12]. Furthermore, feature reduction conducts to less complex decision tree models than instance reduction.

One of the first attempt to inject randomness into the classifiers by altering the feature subset is due to Zheng and Webb [13]. They randomly perturb the sets of attributes available to a C4.5 decision tree induction algorithm at each node looking for more independent classifiers. The authors call this approach Stochastic Attribute Selection Committees (SASC). In a later paper [14], they combine SASC with Boosting to achieve an accuracy higher than either Boosting or SASC alone.

The power of ensemble learning by means of randomized subsets of features has been also shown in [15], where such ensembles are able to improve performances of nearest neighbor classifier ensembles and also of standard methods like Bagging. The improvements are even more impressive because they were achieved without any accuracy ranking of attribute subsets used in voting (all random subset were used).

Bryll et al. [16] develop a method called Attribute Bagging (AB) that combines wrapper feature subset evaluation approach with random subspaces. Before create the ensemble, an “optimal” size for attribute subsets is found by testing classification accuracy of multiple sized random subsets of attributes. Once the size is fixed, randomly selected subsets are evaluated by the learning algorithm and only the better subsets contribute to the ensemble result.

A different approach to create diversity by feature partitioning is to combine features of different levels of relevance. Each resulting subset represents a local maximum of a classification criterion in the feature subset space. Such a method was proposed by Cunnigham and Carney [17], which is also similar in spirit to the genetic algorithm for finding good feature subsets described by Guerra-Salcedo and Whitley [18].

A central problem to previous techniques is the analysis of the relevance of an attribute. It can be stated that irrelevant attributes are not useful for a learning rule, and neither are relevant ones automatically useful. Caruana and Freitag [19] agree that an attribute a is relevant to a learning problem P, if there exists any learner L using any set of attributes A and any distribution of typical training sets, for which the attribute a is useful in the sense of improving L's performance on P when a is added to A. Note that usefulness is measured with respect to a particular learner, using a particular set of attributes, and given a particular distribution of training patterns. According to previous analysis, the following principles can be identified as the main keys to deal with the problem of generate diverse subsets of attributes:

- **dependencies on learner, problem and instances**: the subset should be useful for its particular learner in the ensemble; in other words, the relevance obtained by a filter criterion must be strongly related to the usefulness for its classifier.
- **minimum redundancy**: the redundancy or common information among features in the subset ought to be the lowest possible [20, 21].
- **maximum discrimination capacity**: the criteria should maximize discrimination among classes using class label information, regarding that different subsets of classes are best distinguished by different feature subsets [22, 23].

3.2 Instance set reduction

Both in pattern recognition and machine learning domains, many instance reduction methods have been proposed in the literature (two excellent surveys can be found in [2, 24]). The problem of instance set reduction is primarily related as the deletion from a training set of irrelevant and harmful instances, while retaining only critical instances. By removing a number of instances, the response time for distance-based classification will decrease as fewer instances are examined when a new input vector is presented. Moreover, the removal of noisy instances can also lead to an increase in the corresponding classification accuracy. Such two objectives are generally named *condensing* and *editing*, respectively [25], although other terms have been also employed to define them. For example, *thinning* and *pruning* can be found in the literature as synonyms for condensing, and *filtering* and *cleaning* have been also used to refer to editing. While the condensing algorithms aim at selecting a sufficiently small subset of instances without a significant degradation of classification accuracy, the editing approaches eliminate erroneous instances from the original training set and "clean" possible overlapping among regions from different classes.

In this section, rather than provide a new survey of the field, we briefly report some instance set reduction techniques that make use of the general idea of combining classifiers. Despite the vast amount of instance reduction methods, it is to be remarked that, as far as we know, not many ensembles of these algorithms have been suggested.

One of the first attempts to use an ensemble of classifiers for instance set reduction corresponds to the work by Alpaydin [26]. He proposed to train multiple subsets of the original data set and take a vote over them. When the training set is of moderate size, he uses bootstrapping to generate smaller subsets. When the training set is large, he partitions it into smaller, mutually exclusive subsets and then train the voters.

Brodley and Friedl [27] introduced an innovative approach to identify mislabeled instances in the training set. This consists of employing an ensemble of classifiers to decide which instances are to be considered as atypical and consequently, removed from the original data set. In brief, the idea of Brodley and Friedl is to create an ensemble to process the training set, in such a way that an instance is identified as atypical if h of the m individual filtering schemes cannot classify it correctly.

Following with the idea of using an ensemble of filters to "clean" the input data set, Barandela et al. [28] presented a combining method that simultaneously allows removals and relabeling of training instances: an instance is removed from the training set if less than half of the m individual classifiers coincide in the label assignment; if more than half of the m classifiers agree, then the training instance receives the class label assigned by the majority of the classifiers (this label can be different from the original one).

Very recently, Kim and Oommen [29] have suggested a recursive instance reduction algorithm. Rather than process the whole data set, they propose to recursively subdivide the training set into smaller subsets. After this recursive subdivision, the resulting subsets are reduced with any traditional instance set reduction technique. The resultant sets of instances obtained are, in turn, gathered and processed at the higher level of the recursion to yield more refined instances. This sequence is invoked recursively to ultimately obtain the final reduced set of instances.

4 Discussion and future research

This paper discusses a distributed ensemble of data and classifiers, as a solution to the analysis of huge data sets. Unlike most of ensembles, its main objective is not to improve individual discrimination performances, which is also pursued, but to make feasible the learning of very large data sets. At training stage, this strategy encompasses a partition of data into disjoint moderate-sized subsets, and the independent learning from them by a number of individual classifiers. These self-supported components allows parallel operation over multiple processors. After learning, when an unknown instance need to be classified, the overall prediction is made by combining individual decisions of independent classifiers.

A first issue is how to partition the original data set. Random partitioning is a good alternative because of its computational simplicity and its satisfactory results in general. More intelligent and time-consuming partitioning methods, like clustering, have been examined leading to better learning results [5]. However, their application on huge data sets could be infeasible. Therefore, an obvious choice seems to be a linear process that partition such a huge data set following certain greedy criterion which tends to maximize complementarity of subsets.

Another question is on how to use and combine individual classifiers. About their using, a straight possibility is to employ them to directly learn from disjoint subsets. In [5], experiments on 9 well-known machine learning data sets demonstrate that combined results from disjoint subsets are not better, in general, than a single prediction over the whole data set. It is important to note that none of these 9 datasets is really a huge amount of data. In that same paper, the authors proposed the use of Bagging on subsets to improve their individual performances. The overall prediction was made from Bagging decisions. This two-step combined prediction approach was able to perform better than a single learner over the entire dataset.

Therefore, resampling techniques on moderate-sized sets appear as an effective way to learn from disjoint subsets. Many papers [8, 30, 31] have reported exhaustive analysis and evaluations of both methods considering multiple moderate-sized data sets and some classification models. Some of their conclusions are: i) Bagging consistently produces better accuracies than individual classifiers; ii) Boosting is more irregular, producing significant error reduction for some cases (even compared to Bagging), but sometimes degenerating error rates of single classifiers; iii) Bagging is more resilient to noise than Boosting, which can also overfit in the presence of noise; iv) much of reduction in errors is achieved after 10 or 15 classifiers, and most of reduction, at about 25 classifiers. These results are mainly based on the use of decision trees as individual learners due to their training speed and well-defined parameters setting.

Previous sections talk over about data-reduction-based classifiers, which mostly refer to classifiers using reduced versions of some data set. Reduction techniques generalizes algorithms for features or instances reduction, which look for removing noise and/or redundancy in data. In particular, this paper aims to integrate those data-reduction-based classifiers as individual learners in this distributed ensemble model. In more details, it proposes to use such reduction techniques to obtain smaller and more discriminative versions of the resampled sets of disjoint subsets or, directly, of disjoint subsets. Finally, individual classifiers can take low-cost decisions on reduced data sets.

Many open issues remain. Much of the methods that perturb the training set do not take into account the classification models to be used on such perturbed data. No all distributions or subsets of attributes are adequate for a particular learner in a particular problem. So, an interesting question is, given the classifiers to be combined, how to partition or resample the training set to let them perform better both individually and as a committee.

A related concern is the "optimal" number of partitions or classifiers. Obviously, the error can not be infinitely reduced when increasing the number of learners. Moreover, as this number increases, the probability to find lowly correlated classifiers decreases. In multiprocessing schemes, the number of processors and the memory limitations could be also additional constraints. In cases of disjoint partitions, a preliminary study about how much random information could be removed from the original data without a significant degradation of its discriminant capacity may lead to a useful criterion. These issues may define some of the future lines of research in ensembles of classifiers.

Acknowledgments

This work has been supported in part by grants TIC2003-08496 from the Spanish CICYT and P1-1B2002-07 from Fundació Caixa Castelló - Bancaixa.

References

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

[2] Dasarathy, B.V.: Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamos, CA (1990)

[3] Shipp, C.A., Kuncheva, L.I.: Relationships between combination methods and measures of diversity in combining classifiers. Information fusion **3** (2002) 135–148

[4] Chan, P., Stolfo, S.: Learning arbiter and combiner trees from partitioned data for scaling machine learning. In Usama, F., Uthurusamy, R., eds.: Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD-95), Montreal, Canada, AAAI Press (1995) 39–44

[5] Chawla, N., Eschrich, S., Hall, L.: Creating ensembles of classifiers. In: Proceedings of International Conference on Data Mining (ICDM'01), San Jose, Califormia, IEEE (2001) 580–581

[6] Breiman, L.: Bagging predictors. Technical Report 421, Department of Statistics. University of California, Berkeley. (1994)

[7] Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning. (1996) 148–156

[8] Breiman, L.: Arcing clasifiers. The annals of statistics **26** (1998) 801–849

[9] Freund, Y., Schapire, R.: A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence **14** (1999) 771–780

[10] Tumer, K., Ghosh, J.: Classifier combining: Analytical results and implications. In: Working notes from the Workshop Integrating Multiple Learned Models, Porland (1996)

[11] Dietterich, T.G.: Machine learning research: Four current directions. AI Magazine **18** (1997) 97–136

[12] Skurichina, M., Duin, R.P.W.: Bagging and the random subspace methods for redundant features spaces. In: Multiple Classifier Systems, J. Kittler, F. Roli (eds.), LNCS 2096, Springer-Verlag (2001) 1–10

[13] Zheng, Z., Webb, G.I.: stochastic attribute selection committees. In: Proc. of Australian Joint Conf. on Artificial Intelligence, Los Alamitos, CA, Springer Verlag (1998) 321–332

[14] Zheng, Z., Webb, G.I.: Integrating boosting and stochastic attribute selection committees for further improving the performance of decision tree learning. In: Proc. 10^{th} Int. Conf. on Tools with Artificial Intelligence, Los Alamitos, CA, IEEE Computer Society Press (1998) 216–223

[15] Bay, S.D.: Combining nearest neighbour classifier through multiple feature subsets. In: In. Proc. 17^{th} Int. Conf. on Machine Learning, Madison, WI (1998) 37–45

[16] Bryll, R., Gutierrez-Osuna, R., Quek, R.: Attribute bagging: Improving accuracy of classifier ensembles by using random features subsets. Pattern Recognition **36** (2003) 1291–1302

[17] Cunnigham, P., Carney, P.: Stability problems with artificial neural networks and the ensemble solution. Technical report, Trinity College Dublin, Computer Science, Technical Report: TCD-CS-1999-52 (1999)

[18] Guerra-Salcedo, C., Whitley, D.: Feature selection mechanisms for ensemble creation: A genetic search perspective. In: Data Mining with Evolutionary Algoritms: Reseach Directions. Papers form the AAAI Workshop, Alex A. Freitas (eds.) Technical Report WS-99-06. AAAI Press (1999)

[19] Caruana, R., Freitag, D.: How useful is relevance? In: Working Notes of the AAAI Fall Symposium of Relevance, LA: AAAI Press, New Orleans (1994) 25–29

[20] Studholme, C., Hill, D.L.G., Hawkes, D.J.: Incorporating connected region labelling into automated registration using mutual information. In: Mathematical Methods in biomedical Image Analysis, A.A. Aminin, F.L. Bookstein and D.C. Wilson (eds.), Los Alamitos, CA, IEEE Computer Soc. Press (1996) 23–31

[21] Sotoca, J.M., Pla, F., Klaren, A.C.: Unsupervised band selection for multispectral images using information theory. In: 17^{th} International Conference on Pattern Recognition, Cambridge, UK, (in press) (2004)

[22] Fisher, R.A.: The use of multiple measurement in taxonomic problems. Ann Eugenics **7** (1936) 179–188

[23] Bruzzone, L., Roli, F., Serpico, S.B.: An extension to multiclass cases of the jeffreys-matusita distance. IEEE Trans. of Geoscience and Remote Sensing **33** (1995) 1318–1321

[24] Bezdek, J.C., Kuncheva, L.I.: Nearest prototype classifier design: an experimental study. International Journal of Intelligent Systems **16** (2001) 1445–1473

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

[26] Alpaydin, E.: Voting over multiple condensed nearest neighbors. Artificial Intelligence Review **11** (1997) 115–132

[27] Brodley, C.E., Friedl, M.A.: Identifying mislabeled training data. Journal of Artificial Intelligence Research **11** (1999) 131–167

[28] Barandela, R., Sánchez, J.S., Valdovinos, R.M.: New applications of ensembles of classifiers. Pattern Analysis and Applications **6** (2003) 245–256

[29] Kim, S.W., J., O.B.: Enhancing prototype reduction schemes with recursion: a method applicable for ”large” data sets. IEEE Trans. on Systems, Man, and Cybernetics - Part B **34** (2004) 1384–1397

[30] Opitz, D., Maclin, R.: Popular ensemble methods: an empirical study. Journal of artificial intelligence research **11** (1999) 169–198

[31] Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: bagging, boosting and variants. Machine learning **36** (1999)