Reduction of Large Training Set by Guided Progressive Sampling: Application to Neonatal Intensive Care Data

François Portet[1], Feng Gao[1], Jim Hunter[1] and René Quiniou[2]
[1]Department of Computing Science, University of Aberdeen, Aberdeen AB24 3UE, Scotland, UK
[2]Irisa, INRIA, Campus de Beaulieu, 35042, Rennes, France
{fportet,fgao,jhunter}@csd.abdn.ac.uk, quiniou@irisa.fr

Abstract

Although large training sets are supposed to improve the performance of learning algorithms, there are limits to the volume of data such an algorithm can handle. To overcome this problem, we describe an improvement to a progressive sampling method by guiding the construction of a reduced training set. The application of this method to neonatal intensive care data shows that it is possible to reduce a training set to a third of its original size without decreasing performance.

1. Introduction

Intensive Care Units generate large volumes of data - about 1 MB per patient per day. However, such large volumes are difficult to analyze, so data mining or machine learning techniques are often used to learn classifiers for prediction and decision support. Although the general approach is to learn classifiers from the largest possible dataset, learning a classifier from too large a dataset can be computationally impossible or time-consuming and thus the training set must be reduced.

'Data reduction' aims at aggregating the information contained in large datasets into manageable smaller information chunks, using simple tabulation, clustering, principal component analysis (PCA), etc. However, these methods need either data pre-processing or modification of the example datasets in such a way that it is more difficult to interpret the model which has been learned (e.g. PCA). Progressive Sampling (PS) [Provost et al. 1999] incrementally constructs a training set from a larger dataset without decreasing the classification performance and without altering the initial format of the examples. In this paper, we propose a variant of PS and show its application to the domain of Neonatal Intensive Care.

2. Progressive Sampling

Progressive Sampling (PS) starts with a small training subset (*TS*) of the full dataset (*FDS*) and incrementally extends *TS* until the learning accuracy satisfies some convergence criteria. The resulting dataset is expected to be smaller than *FDS* and to lead to (at least) the same performance. Figure 1 shows the general algorithm.

Let *FDS* be the Full Dataset
Let $S = \{n_0, .. n_k, \ldots n_K\}$ be the planned sizes of *TS*
$k = 0$;
While *not converged* **do**
  TS ← computeTS(*FDS*) // copy n_k examples from *FDS* to *TS*
  M ← learn(*TS*) // learn the model *M*
  Evaluate(*M, FDS*) // evaluate *M* on *FDS*
  inc(*k*)
End do
Return *M*

Fig. 1 Progressive sampling algorithm.

Before starting the learning process, the progressive sizes of *TS* are scheduled (planned). Then *TS* is used to learn the classifier model *M* (by a decision tree, neural network, etc.) which is tested until convergence is attained. The optimal training set is computed by mean of a learning curve which is used to retain the best balance between size and learning performance. Provost *et al.* [1999] have showed that when dealing with large volume of data, PS is more efficient than using the entire dataset. However, PS does not explicitly deal with unbalanced datasets. To face this problem, Ng and Dash [2006] introduced a method to improve the relative distribution of each class by over-sampling the minor class in *computeTS(FDS)*. But, as they emphasized, replicating examples from the smaller classes (over-sampling) leads to over-fitting.

These approaches select the examples to be added into *TS* at random. We believe that it is possible to speed up the convergence by using *a priori* information to select the most appropriate examples to add.

3. Guided Progressive Sampling

Guided PS (GPS) uses a distance measure *d* between the samples in *TS* and the samples in *FDS* to guide the selection of samples to add to *TS*. Once *M* is learned, each $e_i \in FDS$ is tested to form the triple $(e_i, m(e_i), d(e_i))$ where $m(e_i)$ is the

result of the classification of e_i using M ($m(e_i) \in$ {correct, incorrect}) and $d(e_i)$ gives the distance from e_i to the centroid of the class to which it actually belongs. This set of triples is used in *computeTS(FDS)* according to one of two strategies:

1. ***GPS*** adds to each class in *TS*, the worst *misclassified* examples i.e. those with the highest values of $d(e_i)$. This is intended to improve learning robustness by considering the difficult cases.
2. ***GPS+*** extends GPS by additionally adding the best *correctly classified* examples i.e. with the lowest values of $d(e_i)$. This is intended to reinforce learning stability which can be distorted by only including the worst misclassifications.

These choices rest on the assumption that learning is most influenced by the extreme examples of each class (correct classifications and misclassifications). The distance measure d does not need to be exact (otherwise it would be directly used to learn the model!) but is a heuristic estimate of how much the classification is wrong.

4. Case study: bradycardia detection

The method has been tested on the detection of bradycardias by decision tree learning (C4.5 with pruning). The dataset consists of thirteen heart rate (HR) time series each covering 24-hours recorded from premature babies receiving intensive care. The episodes of bradycardia were annotated by two clinical experts. Each example in FDS is described by 25 attributes (raw HR value and min, max, slope etc. over several centered windows). The size of the complete 13 record dataset is more than 80MB. Given such a large dataset, learning on the entire set is impossible. Moreover, the dataset is completely unbalanced. For example, in record #16234, the *bradycardia* class contains 533 examples whereas the *no-bradycardia* class contains 79875 examples, the *bradycardias* representing only 0.66% of the total dataset. However, this is to be expected, as bradycardia is defined as a short transient event. In addition, the records contain episodes of artifact that can perturb learning.

Random sampling (RS), GPS, and GPS+ have been used. To try to balance the large difference between *bradycardia* and *no-bradycardia*, the initial TS contained 100% of the *bradycardias* and 1% (selected at random) of the *no-bradycardias*. On each iteration, 3.33% more of the *no-bradycardias* were selected from *FDS* according to the particular strategy in use. Learning was stopped when the learning curve become sufficiently stable [Provost *et al.,* 1999].

Fig.2 shows the number of classification errors against the training size for record #16234. GPS converged faster than GPS+ and RS. GPS+ and RS converged at the same iteration however GPS+ led to higher accuracy. GPS and GPS+ produced more errors at the beginning of the process as they initially selected the most difficult examples to classify but this led rapidly to a more stable plateau (fewer oscillations) than RS. The figure shows that after reaching the beginning of the plateau, the examples added do not provide information that has not already been learned by the decision tree. Results found with GPS led to 111 errors for TS=9317 examples. Thus around 90% of the dataset is not useful for learning. The decision tree learned with GPS led to the same performance (111 errors) as the decision tree learned from FDS but with a slightly smaller tree. The proportion of *bradycardias* is still not equally distributed but increased from 0.66% to 5.72%.

Mean accuracy over the 13 datasets was 99.66% for RS (20 runs), 99.84% for GPS+ and 99.85% for GPS, with significant differences between GPS (or GPS+) and RS ($p<0.04$ in the worst case, $p<0.0001$ for #16234).

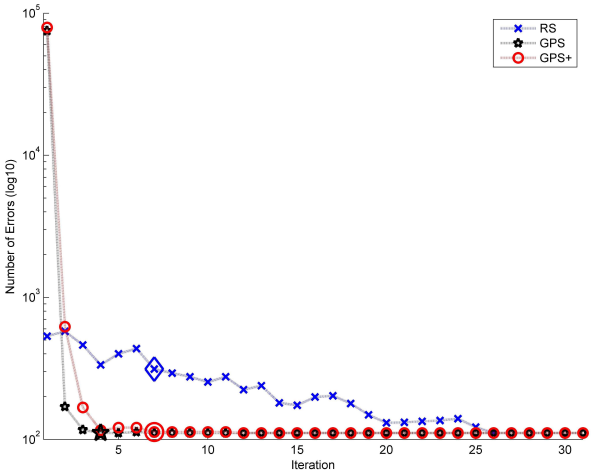

Fig. 2 Progressive learning for the record #16234. Large marks show convergence.

5. Discussion

Guided progressive sampling has shown to be more efficient than random progressive sampling for learning from a massive training set. Using *a priori* knowledge to guide the sampling leads to a faster convergence and a better selection of the "relevant" examples to use for learning. Further experiments will be undertaken to improve bradycardia detection with the reduction of larger datasets. This approach can also be useful in a situation with a small dataset to capture the "best" training examples.

References

[Provost *et al.,* 1999] F. Provost, D. Jensen and T. Oates. Efficient progressive sampling. In *Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Diego, USA. 1999.

[Ng and Dash, 2006] W. Ng and M. Dash. An Evaluation of Progressive Sampling for Imbalanced Datasets. In *Sixth IEEE International Conference on Data Mining Workshops*. Hong Kong, China. 2006.