Ensembles of Classifiers from Spatially Disjoint Data

Robert E. Banfield[1], Lawrence O. Hall[1], Kevin W. Bowyer[2], W. Philip Kegelmeyer[3]

[1]Department of Computer Science and Engineering, ENB118
University of South Florida
4202 E. Fowler Avenue
Tampa, FL 33620-9951, USA
{rbanfiel, hall}@csee.usf.edu
[2]Department of Computer Science and Engineering
University of Notre Dame
South Bend, IN 46556, USA
kwb@cse.nd.edu
[3]Sandia National Laboratories
Biosystems Research Department
P.O. Box 969, MS 9951
Livermore, CA 94551-0969, USA
wpk@ca.sandia.gov

Abstract. We describe an ensemble learning approach that accurately learns from data which has been partitioned according to the arbitrary spatial requirements of a large-scale simulation wherein classifiers may be trained only the data local to a given partition. As a result, the class statistics can vary from partition to partition; some classes may even be missing from some partitions. In order to learn from such data, we combine a fast ensemble learning algorithm with Bayesian decision theory to generate an accurate working model of the simulation. Results from a simulation of an impactor bar crushing a storage canister and from region recognition in face images show that regions of interest are successfully identified.

1 Introduction

We consider the problem of dealing with an amount of data too large to fit in the memory of any one computer node and too bandwidth intensive to move around to neighboring nodes, a problem which has far reaching implications [1]. Since the data cannot be moved around between nodes, there may exist no logical grouping other than the order in which it was originally stored. Such a problem exists for the United States Department of Energy's Advanced Simulation and Computing program [2], wherein a supercomputer simulates a hypothetical real-world event. Data is stored on disks attached to compute nodes according to its spatial location within the 3D simulation. The concern is that the storage allocation for the simulation optimizes for balanced and efficient computation, without regard to conditions that might make it easy or difficult for a machine learning algorithm to use the resulting data.

In analyzing these simulations, developers and users want to spot anomalies which may take days or weeks to find in a massive simulation. So, marking some areas of interest and finding others in the same or similar types of simulations can greatly reduce the time to debug and analyze a simulation. Generally, experts will manually designate salient areas in the simulation as "interesting" according to personal, subjective criteria. This process would be markedly sped up by analyzing those points and suggesting new points across each compute node.

In this paper, we show examples from a simulation of a storage canister being crushed by an impactor bar from above at approximately 300 miles per hour. In order to illustrate how the complete simulation appears, a visualization of the partitions is provided below in Figure 1. The different shades of grey represent the partitioning of the simulation in a distributed environment. Note that pieces of the impactor bar crushing the canister are also broken up spatially according to the partition.

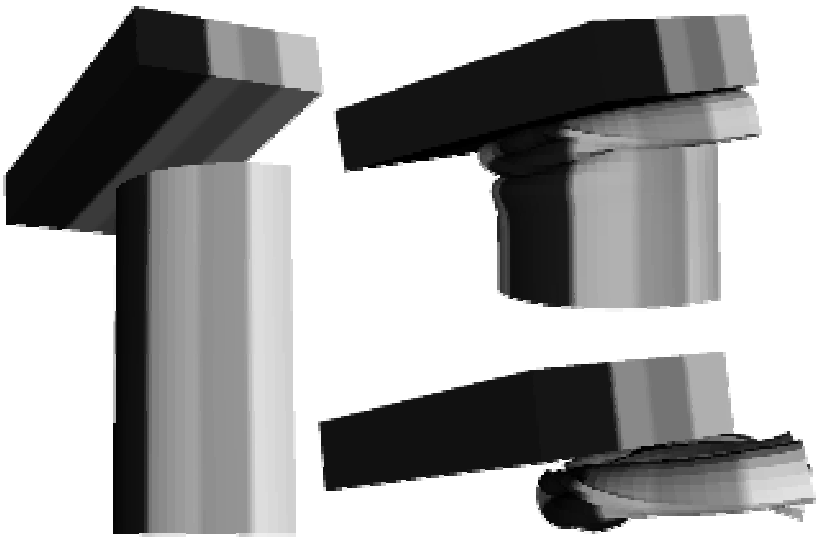

Fig. 1. A visualization of the data as distributed across compute nodes. There are four partitions shown in different gray levels as the storage canister is crushed.

As a result of the partitioning, areas of saliency may be limited to only a few nodes. Salient points, being few in number, exhibit a pathological minority class classification problem. In the case of a partition having zero salient points, a single-class "classifier" will be learned. Furthermore, a prominent event on one node is not necessarily indicative of saliency on another node experiencing a similar event.

We show that it is possible to obtain an accurate prediction of salient points even when the data is broken up arbitrarily in 3D space with no particular relation to feature space. Results on this data set indicate that experts working with much larger simulations can benefit from the predictive guidance obtained from only a small amount of relevant data.

2 Data description

In this paper we look at experiments in which a canister is rapidly crushed much like a person might crush a soda can. The walls of the canister buckle under the pressure and top of the canister accelerates downward until it meets the bottom. In our experiments we observe 44 slices of time in which the above event is simulated and recorded.

2.1 Physical and Spatial Characteristics

Ten physical variables are stored for each of 10,088 nodes within each of 44 time steps. They are the displacement on the X, Y, and Z axes; velocity on the X, Y, and Z axes; acceleration on the X, Y, and Z axes; and "Equivalent Plastic Strain" which is a metric for the stress on the surface of the canister [3]. The total number of data samples is 44•10088=443,872.

The data for each of the time steps is divided spatially according to the compute node to which it is assigned. The partitioning is performed vertically along the Y axis of the canister, dividing the canister into 4 disjoint spatial partitions of roughly equal size. Each compute node can see only one of these partitions, and we stipulate that it is too expensive in time or storage space to move data to another compute node.

2.2 Train and Test Sets

For every time step, those pieces of the canister that have buckled and been crushed are marked as salient. At the beginning of the simulation, before the impactor bar has made contact, there are no salient nodes within the mesh. As time progresses and the canister collapses, more and more nodes are marked salient.

The process of marking salient nodes within the mesh can be as precise as the expert demands. However, a high level of precision requires a correspondingly high level of effort marking the data. In order to model a practical scenario where an expert is more interested in saving time than catering to the nuances of Machine Learning, we have included a fair amount of noise in the class labels by using tools which mark areas as salient rather than individual points—there are over 10,000 points per time step. Since the impactor bar and the canister are so close in proximity, it is quite reasonable to assume the bar will often have areas incorrectly marked as salient.

In each time step and in each partition, saliency is designated in the above fashion. For each partition, data present in the time steps are collapsed into two segments, a training set and a test set, according to the time step number: even time steps are combined into a training set, odd time steps are combined into a test set. Therefore our experiments utilize four partitions each having two data sets, for a total of eight data sets.

3 Classification system

For each training set developed on each compute node, we utilized Breiman's random forest algorithm to rapidly generate an ensemble of classifiers. The motivation for using this ensemble technique stems from the inherent speed benefit of analyzing only a few possible attributes from which a test is selected at an internal tree node. A complete description of the random forest algorithm can be found in [4]. Its accuracy was evaluated in [5] and shown to be comparable with or better than other ensemble generation techniques.

Classification of a test point within the simulation involves prediction by each partition's random forest. Because our algorithm is designed to work when only a few compute nodes have salient examples, a simple majority vote algorithm may fail to classify any salient points if the number of compute nodes trained with salient examples is less than half of the number of compute nodes. In a large-scale simulation it is very likely that there will be nodes which have no salient examples in training. Therefore we must consider the *priors*: the probability that any given node contained salient examples and therefore is capable of predicting an example as salient. A breakdown of our algorithm follows.

$p(w_1|x)$ = number of ensembles voting for class w1 for example x,
$P(w_1)$ = number of ensembles capable of predicting class w1
Classify as w_1 if: $p(w_1|x)/P(w_1) > p(w_2|x)/P(w_2)$
Classify as w_2 if: $p(w_1|x)/P(w_1) < p(w_2|x)/P(w_2)$

Of course, this is nothing more than Bayesian decision theory applied to the majority vote for a two class problem. Moving to an n-class problem is trivial:

Classify as w_n: $\text{argmax}_n \, (p(w_n|x)/P(w_n))$

4 Experiments

The random forest of each partition returns a single prediction for a class. Those predictions are combined into a single prediction for the example as outlined above using the Bayesian majority vote with priors. Training is performed on the data contained in the even time steps. Predictions on odd time steps are compared to the marked saliency in the odd time steps to obtain an estimate of the true error. We also obtain predictions on even time steps to model the resubstitution error.

Fig. 2. *Left*: Ground truth as labeled in time step 3. *Right*: Predicted class labels.

Fig. 3. *Left*: Ground truth as labeled in time step 19. *Right*: Predicted class labels.

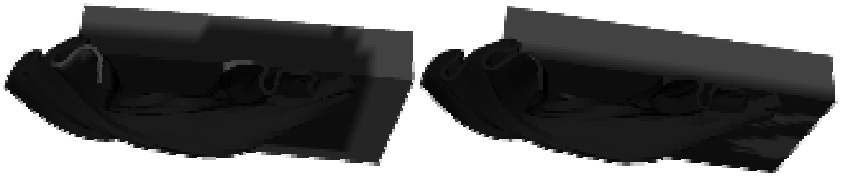

Fig. 4. *Left*: Ground truth as labeled in time step 37. *Right*: Predicted class labels.

Fig. 5. *Left*: Training data as labeled in time step 4. *Right*: Predicted class labels.

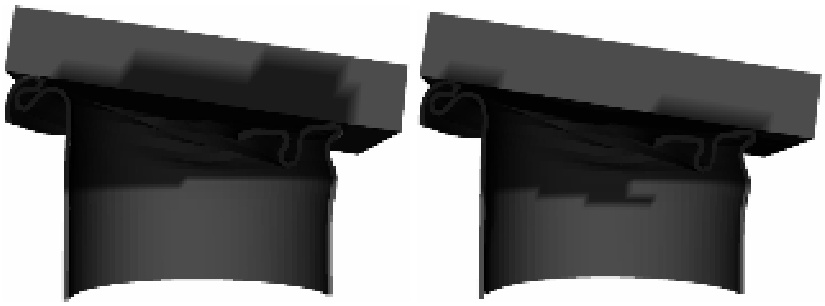

Fig. 6. *Left*: Training data as labeled in time step 20. *Right*: Predicted class labels.

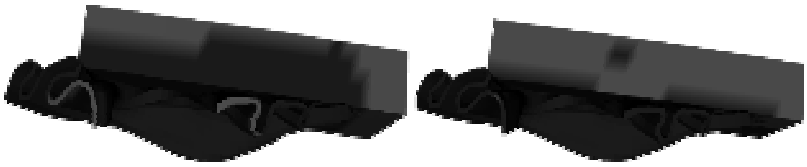

Fig. 7. *Left*: Training data as labeled in time step 38. *Right*: Predicted class labels.

5 Results

The goal of the prediction stage is to direct experts to additional salient regions. Unfortunately, a metric for determining how accurate an algorithm is in finding and classifying regions is non-trivial. For this reason, we provide figures to help illustrate the accuracy of our approach. Figures 2, 3, and 4 show the algorithm's predictive power as the canister is still in the process of being crushed. In Figures 5, 6, and 7 we observe the resubstitution error on the training data. Darker areas indicate regions which have been classified as salient. Ensemble predictions are provided to the right of the labeled data in each of the figures.

For conventional reasons, we provide an estimate of the true error being 25.4% using the methods outlined above. Because this error is based on a point to point comparison between the test set and the predictions upon the test set, and because we know "regions" are salient rather than "points," we could potentially lower the error by utilizing image processing techniques such as erosion and dilation.

6 Previous Work Revisited

In revisiting our previous work [1], we witness definite improvements in the classification accuracy of face images obtained from the FERET database [6] using the approach discussed here. In those experiments we utilized a k-nearest centroid algorithm which lacked adequate speed for terascale data sets but achieved reasonable results given our assumption of spatially disjoint subsets. An example from the database is shown below in Figure 8.

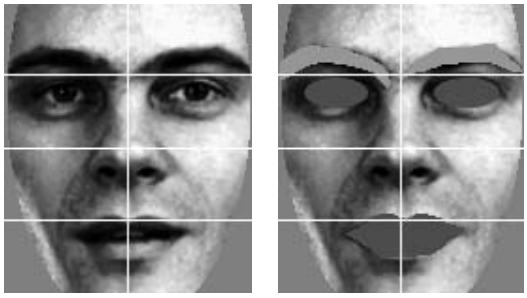

Fig. 8. Image from the FERET database showing marked saliency for both "Interesting" and "Somewhat Interesting" classes for eight partitions delineated by white lines. The "Interesting" class contains the eyes and mouth. The "Somewhat Interesting" class contains only the eyebrows.

In previous experimentation with a k-nearest centroid algorithm we were able to identify salient regions, however regions of noise were also labeled. These experiments did not use the Bayesian majority vote. We compare those results with a forest of 1000 random forest trees trained on each of the eight partitions in

combination with Bayesian majority voting using priors. Many fewer pixels are labeled incorrectly using this later method. A comparison of the k-nearest centroid algorithm using 11 centroids to eight random forests of 1000 decision trees is shown in Figure 9. Neither provide for significant differentiation between the "interesting" and "somewhat interesting" classes due to the weakness of the derived feature attributes.

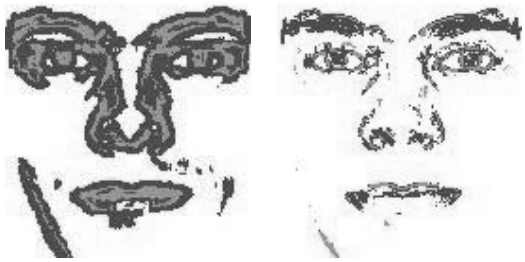

Fig. 9. *Left*: Saliency predictions using 11 centroids. *Right*: Bayesian majority vote with priors using 1000 random forest trees per partition.

7 Summary and Discussion

Some simulations must be broken up across multiple processors in order to obtain results in a reasonable amount of time. The method of breaking data into pieces is not necessarily valuable, and possibly even harmful, to machine learning algorithms, as it violates the usual assumption that class statistics will be the same across all the training data and the test data. In this paper we have shown how large simulation data broken up non-intuitively into spatial regions may be classified using a combination of fast ensemble techniques and Bayesian decision theory.

Our preliminary results on a relatively small problem indicate that our approach has merit. In our simulation of the crushing of the storage canister, the resultant predictions appear more accurately classified than the training data which has been labeled haphazardly in accordance with time constraints placed upon experts. This may signify that the algorithm learning the underlying function which determines which points are salient, with the overlap of uninteresting points outweighing the very large number of uninteresting points overall. A comparison with our previous work using facial data also showed improvement.

In preparation for larger simulations with greater minority class problems, we conjecture that we might assign a bias, or risk ($R_n(w_n|x)$), to a particular class utilizing the same sound Bayesian theories off which we based our algorithm:

$$\text{Classify as } w_n\text{: } \operatorname{argmax}_n \left(R_n(w_n|x)/P(w_n)\right) .$$

It may also be possible to assign dynamic weights to the classifiers as shown in [7].

We believe the speed associated with the rapid generation of ensemble classifiers will enable the tractable prediction of saliency in much larger data sets. The general problem of creating an ensemble from data that was partitioned without regard to the simplicity of the machine learning algorithm is an important practical problem that merits additional attention.

Acknowledgments

This research was partially supported by the Department of Energy through the Advanced Strategic Computing Initiative (ASCI) Visual Interactive Environment for Weapons Simulation (VIEWS) Data Discovery Program Contract number: DE-AC04-76DO00789.

Bibliography

1. Lawrence O. Hall, Divya Bhadoria, and Kevin W. Bowyer. "Learning a model from spatially disjoint data," 2004 IEEE International Conference on Systems, Man and Cybernetics, October 2004.
2. National Nuclear Security Administration in collaboration with Sandia, Lawrence Livermore, and Los Alamos National Laboratories, "http://www.sandia.gov/NNSA/ASC/"
3. B.S. Lee, R.R. Snapp, R. Musick, "Toward a query language on simulation mesh data: an object oriented approach," Proceedings of the International Conference on Database Systems for Advanced Applications, Hong Kong, April 2001.
4. L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5-32, 2001.
5. Robert E. Banfield, Lawrence O. Hall, Kevin W. Bowyer, Divya Bhadoria, W. Philip Kegelmeyer and Steven Eschrich, "A comparison of ensemble creation techniques", The Fifth International Conference on Multiple Classifier Systems, Cagliari, Italy, June, 2004.
6. "The facial recognition technology (FERET) Database", http://www.itl.nist.gov/iad/humanid/feret/
7. Michael Muhlbaier, Apostolos Topalis, and Robi Polikar, "Learn++.MT: A new approach to incremental learning," The Fifth International Conference on Multiple Classifier Systems, Cagliari, Italy, June, 2004.