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 that has been partitioned according to the arbitrary spatial requirements of a large-scale simulation wherein classifiers may be trained only on 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 predictive model of the simulation data. 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 that 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 to find in a massive simulation, especially if it is important to spot every anomaly. 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 (examples) 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.

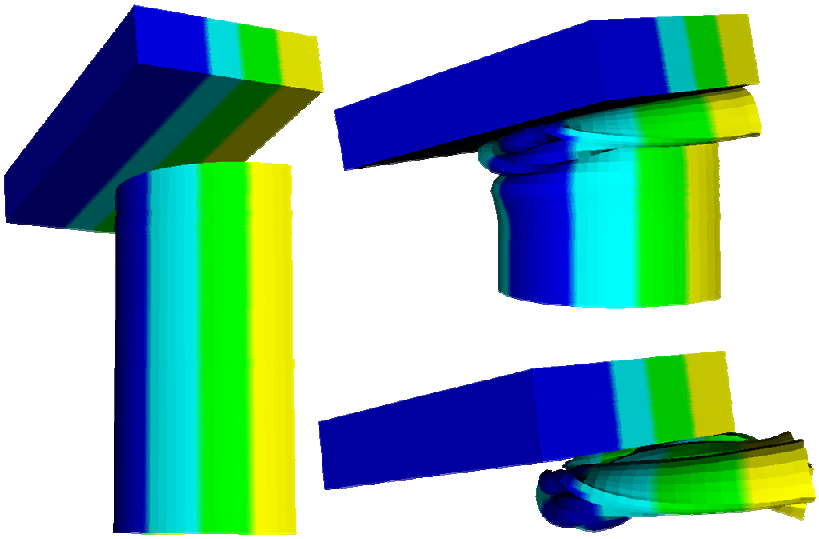

Fig. 1. A visualization of the data as distributed across compute nodes. There are four partitions shown in different gray levels as the storage canister is crushed.

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 the 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 was simulated and recorded.

2.1 Physical and Spatial Characteristics

Nine physical variables were 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; and acceleration on the X, Y, and Z axes. The total number of data samples was 44•10088=443,872.

The data for each of the time steps was divided spatially according to the compute node to which it is assigned. The partitioning was performed vertically along the Y axis of the canister, dividing the canister into four disjoint spatial partitions of roughly equal size. Each compute node can see only one of these partitions, and we assume 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 were marked as salient. To assist in labeling, the "Equivalent Plastic Strain" (EQPS), a measure for the stress on the surface of the canister [3], has been calculated and was used as a general template in choosing salient points. At the beginning of the simulation, before the impactor bar has made contact, there were no salient nodes within the mesh. As time progresses and the canister collapses, more and more nodes were 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 allowed a fair amount of noise in the class labels by using tools that 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 was designated in the above fashion. For each partition, data present in the time steps was collapsed into two segments, a training set and a test set, according to the time step number: even time

steps were combined into a training set, odd time steps were combined into a test set. Therefore our experiments used four partitions each having two data sets.

3 Classification system

For each training set developed on each compute node, we used Breiman's random forest algorithm to rapidly generate an ensemble of 25 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 was 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 the number of compute nodes. In a large-scale simulation it is 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 during training and so 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: $p(w_1|x)/P(w_1) < p(w_2|x)/P(w_2)$
A tie, $p(w_1|x)/P(w_1) = p(w_2|x)/P(w_2)$, is broken randomly

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:

$$\text{Classify as } w_n\text{: } \operatorname{argmax}_n \left(p(w_n|x)/P(w_n)\right) . \qquad \textbf{(1)}$$

4 Experiments

The random forest of 25 trees for each partition returns a single prediction for a class. Those classifier predictions are combined into a single ensemble 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 on a point-by-point basis to obtain an estimate of the true error. We also obtain predictions on even time steps to evaluate the re-substitution error. Our results are compared with those of using a single classifier within each partition and a single classifier for the entire simulation.

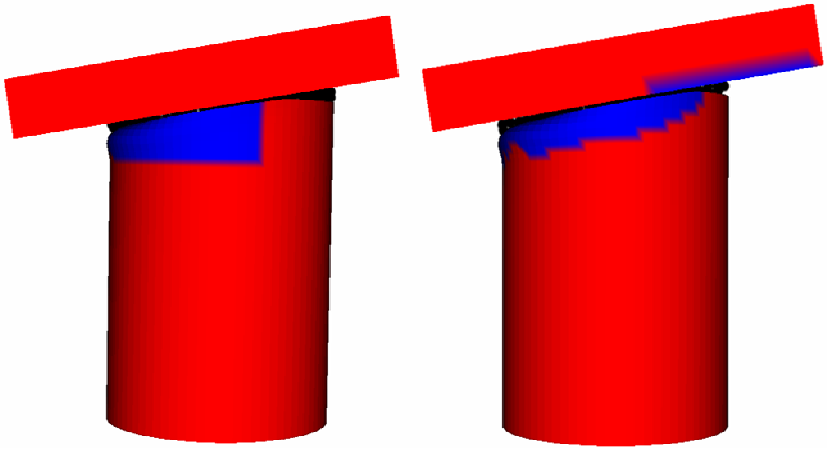

Fig. 2. *Left*: Ground truth as labeled in time step 3. *Right*: Predicted class labels.

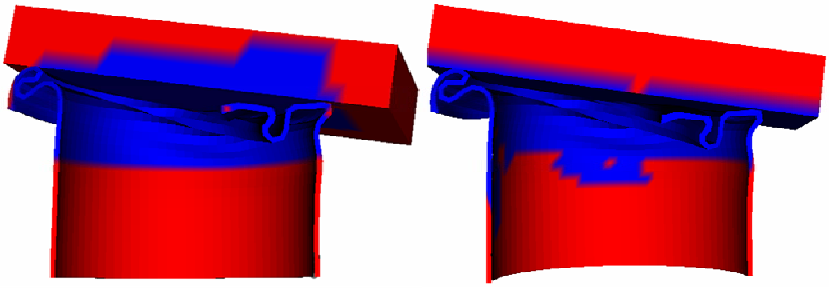

Fig. 3. *Left*: Ground truth as labeled in time step 19. *Right*: Predicted class labels.

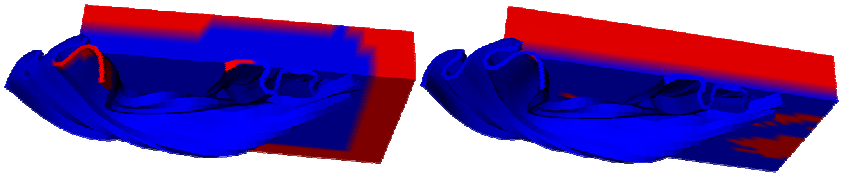

Fig. 4. *Left*: Ground truth as labeled in time step 37. *Right*: Predicted class labels.

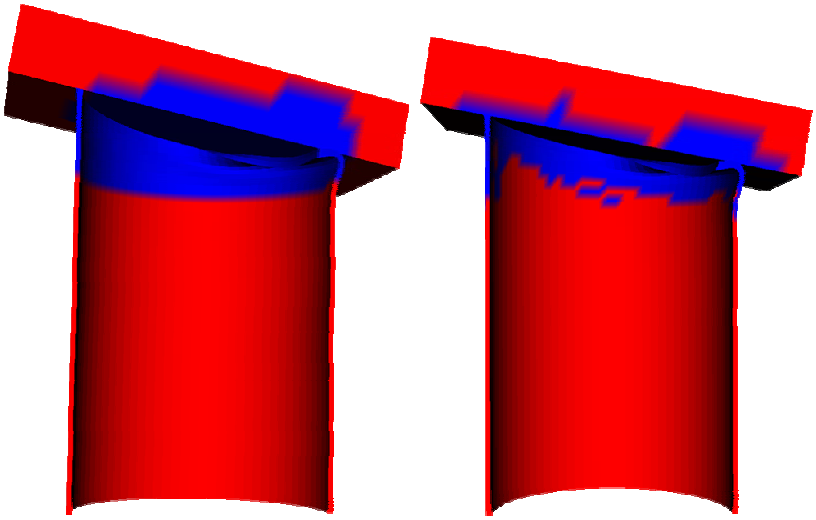

Fig. 5. *Left*: Training data as labeled in time step 4. *Right*: Predicted class labels.

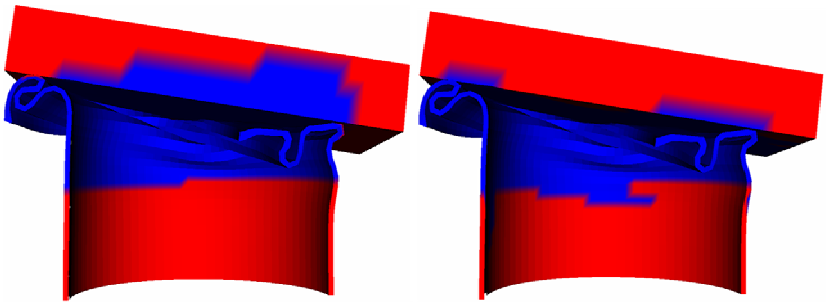

Fig. 6. *Left*: Training data as labeled in time step 20. *Right*: Predicted class labels.

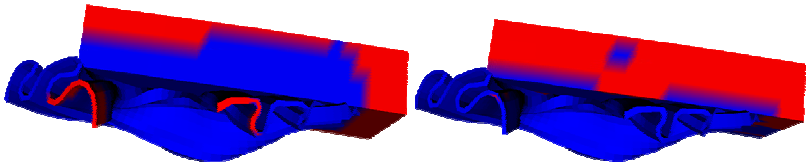

Fig. 7. *Left*: Training data as labeled in time step 38. *Right*: Predicted class labels.

Table 1. Error percentage and m-estimates obtained in our experiments

Algorithm Used	Error Percentage	m-Estimate
Single Classifier	24.3%	0.59
Single Classifier in each Partition	26.7%	0.54
Random Forest in each Partition	25.9%	0.55

5 Results

The goal of the prediction stage is to direct experts towards additional salient regions. Unfortunately, a suitable metric for the algorithm's usefulness 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 while the canister is being crushed. In Figures 5, 6, and 7 we observe the re-substitution error on the training data. Darker areas indicate regions that have been classified as salient. Ensemble predictions are provided to the right of the labeled data in each of the figures.

In Table 1, we provide an estimate of the true error for our experiments. Because this error is based on a point-to-point comparison between our labeled 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. The error rate for a single classifier trained on each partition is 26.7%. The ensemble error rate of 25.9% was not reduced by using 250 classifiers per partition instead of 25.

As a way of calculating how accurate the algorithm was for the minority class we used the m-estimate [6] shown below:

$$P_m=(TP+mb)/(TP+FP+m)\ . \qquad \textbf{(2)}$$

In this equation b is the prior for the minority class, m is the parameter for controlling the shift towards b, and TP and FP represent number of True Positives and False Positives. The prior for the minority class in our problem is 0.30. As suggested in [7], we have chosen m such that bm=10. An evaluation of this method as it corresponds to decision trees is shown in [8]. In the preceding experiments, the forest of trees produced a slightly higher m-estimate, 0.55, than the single tree per partition, 0.54.

Partitioning the data spatially, while necessary for our large simulation, negatively affects the accuracy of the results. In a comparison with the performance of a single pruned decision tree, our multiple classifier spatially partitioned result is 1.6% less accurate. The m-estimate of the single pruned decision tree was 0.59. While such losses in accuracy are unfortunate, this result shows that the problem of having a spatially partitioned data set is non-trivial; even an ensemble created from a random disjoint partitioning often provides an increase in accuracy over a single classifier, especially so in the case of a large dataset [9].

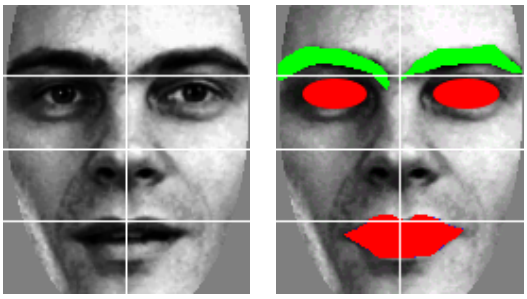

Fig. 8. Image from the FERET database showing marked saliency for both "Interesting" and "Somewhat Interesting" classes for eight partitions delineated by white lines. The "Interesting" class contains the eyes and mouth. The "Somewhat Interesting" class contains the eyebrows.

Fig. 9. *Left*: Saliency predictions using KNC with 11 centroids. *Right*: Bayesian majority vote with priors using 1,000 random forest trees per partition.

6 Previous Work Revisited

In revisiting our previous work [1], we saw definite improvements in the classification accuracy of face images obtained from the FERET database [10] using the approach discussed here. In those experiments we employed 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 in Figure 8.

In previous experimentation with a k-nearest centroid algorithm we were able to identify salient regions though regions of noise were also labeled. These experiments did not use the Bayesian majority vote. We compare those results with a forest of 1,000 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 eleven centroids to eight random forests of 1,000 decision trees is shown in Figure 9. Though the random forest results are 5.2% more accurate than the KNC results for this image, neither provides for significant differentiation between the "interesting" and "somewhat interesting" classes, due to the weakness of the derived feature attributes. The random forest image contains fewer false positive regions than the one created with KNC, especially in the regions to the left and right of the mouth. Fewer false positives would guide researchers examining the data to fewer unimportant regions.

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 (according to a machine learning perspective) into spatial regions may be classified using a combination of fast ensemble techniques and Bayesian decision theory.

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 is learning the underlying EQPS function that was generally used to label salient points. A comparison with our previous work using facial images 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 upon which we based our algorithm:

$$\text{Classify as } w_n\text{: } \operatorname{argmax}_n \left(R_n(w_n|x)/P(w_n)\right) . \qquad \textbf{(3)}$$

It may also be possible to assign dynamic weights to the classifiers as shown in [11].

Of particular interest is the ability to classify regions rather than points since researchers examining these simulations will be looking at areas of interest within the simulation. An algorithm to perform such a task is currently being developed.

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. L. O. Hall, D. Bhadoria, and K. 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. R. E. Banfield, L. O. Hall, K. W. Bowyer, D. Bhadoria, W. P. Kegelmeyer and S. Eschrich, "A comparison of ensemble creation techniques", The Fifth International Conference on Multiple Classifier Systems, Cagliari, Italy, June, 2004.
6. J. Cussens, "Bayes and pseudo-Bayes estimates of conditional probabilities and their reliabilities", Proceedings of the European Conference on Machine Learning, 1993.
7. B. Zadrozny and C. Elkin, "Learning and making decisions when costs and probabilities are both unknown", Proceedings of the Seventh International Conference of Knowledge Discovery and Data Mining, 2001.
8. N. V. Chawla, "C4.5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure, Workshop on Learning from Imbalanced Data Sets II, 2003.
9. N.V. Chawla, T.E. Moore, Jr., L.O. Hall, K.W. Bowyer, W.P. Kegelmeyer and C. Springer, "Distributed learning with bagging-like performance", Pattern Recognition Letters, Vol. 24 (1-3) pp. 455-471, 2003.
10. "The facial recognition technology (FERET) Database", http://www.itl.nist.gov/iad/humanid/feret/
11. M. Muhlbaier, A. Topalis, and R. Polikar, "Learn++.MT: A new approach to incremental learning," The Fifth International Conference on Multiple Classifier Systems, Cagliari, Italy, June, 2004.