Hybrid Approaches for Classifying Parallel Computation

Sean Whalen, Sean Peisert, and Matt Bishop

Abstract

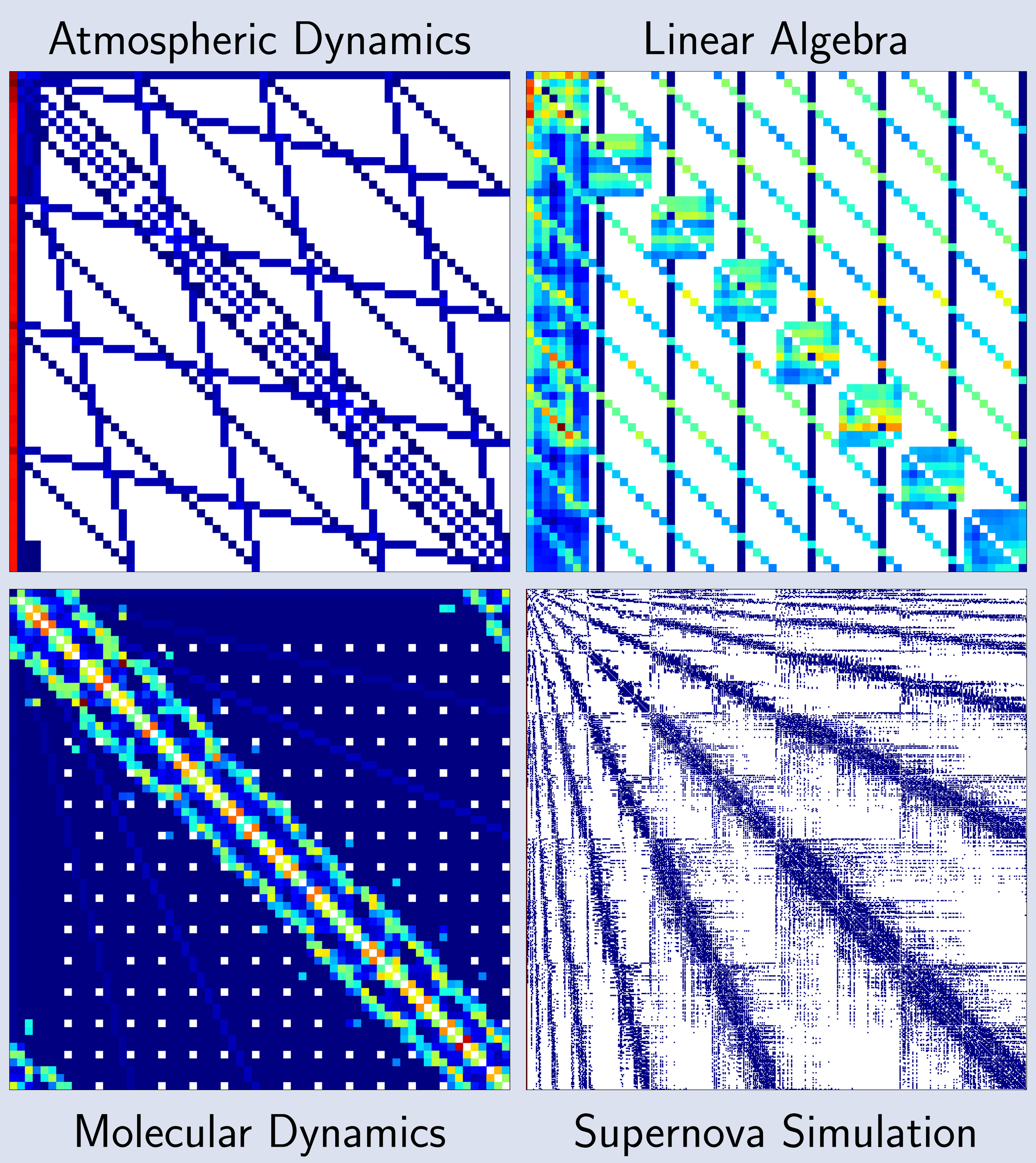
Parallel computation in a high performance computing environment can be characterized by the distributed memory access patterns of the underlying algorithm. During execution, networks of compute nodes exchange messages that indirectly exhibit these access patterns. Identifying the algorithm underlying these observable messages is the problem of latent class analysis over information flows in a computational network. Towards this end, our work applies methods from graph theory, network theory, and machine learning to classify parallel computations solely from network communication patterns. Pattern classification has applications to several areas including anomaly detection, performance analysis, and automated algorithm replacement.

Communication Patterns

Message Passing Interface (MPI) is a standard for distributed memory parallel programs. Our data consists of 5 dimensional vectors of MPI communication features:

[source, destination, call name, bytes sent, repeat count]

Scientific applications have highly structured MPI communications; these are tied closely to their distributed memory access patterns.



These patterns are often robust to changes in architecture and the number of compute nodes, but may vary with different parameters or datasets.

Call Distributions

We first create empirical probability distributions of MPI calls made by each node in two unknown parallel computations, then compare the distributions for corresponding nodes. If some threshold of node pairs match, the computations are deemed the same. We use the two-sample Kolmogorov-Smirnov (KS) test to compare distributions, first computing the *D-statistic* for two empirical cumulative distribution functions:

$$D_{m,n} = \max_x |\hat{S}_m(x) - \hat{S}_n(x)|$$

where m and n are the total event counts of their respective distributions. We then compute the probability that differences in the distributions are due to chance (the *p-value*) and treat the distributions as different if this value is less than our threshold α:

$$P(D_{m,n} \geq D_O) < \alpha$$

for the observed statistic D_O.

Network Motifs

Another approach to characterizing communication topologies is to describe global communication patterns in terms of their localized subgraphs. Those subgraphs that occur more often than would be expected in randomized networks are called *motifs*. Over-representation of a subgraph is determined by its z-score:

$$z = \frac{N_O - N_R}{\sigma}$$

where N_O is its count in the original network, N_R is its mean count in randomized networks, and σ is the standard deviation from N_R.

We create graphs from communication patterns where edges exist between nodes that exchange messages; edges are "colored" with MPI features such as the call name. Single color graphs assign all MPI calls to the same group, while 2-color graphs distinguish between broadcast and point-to-point calls. Lastly, 3-color graphs divide point-to-point calls into send and receive groups. We identified 1-3 color motifs of size 3 and 1-2 color motifs of size 4:

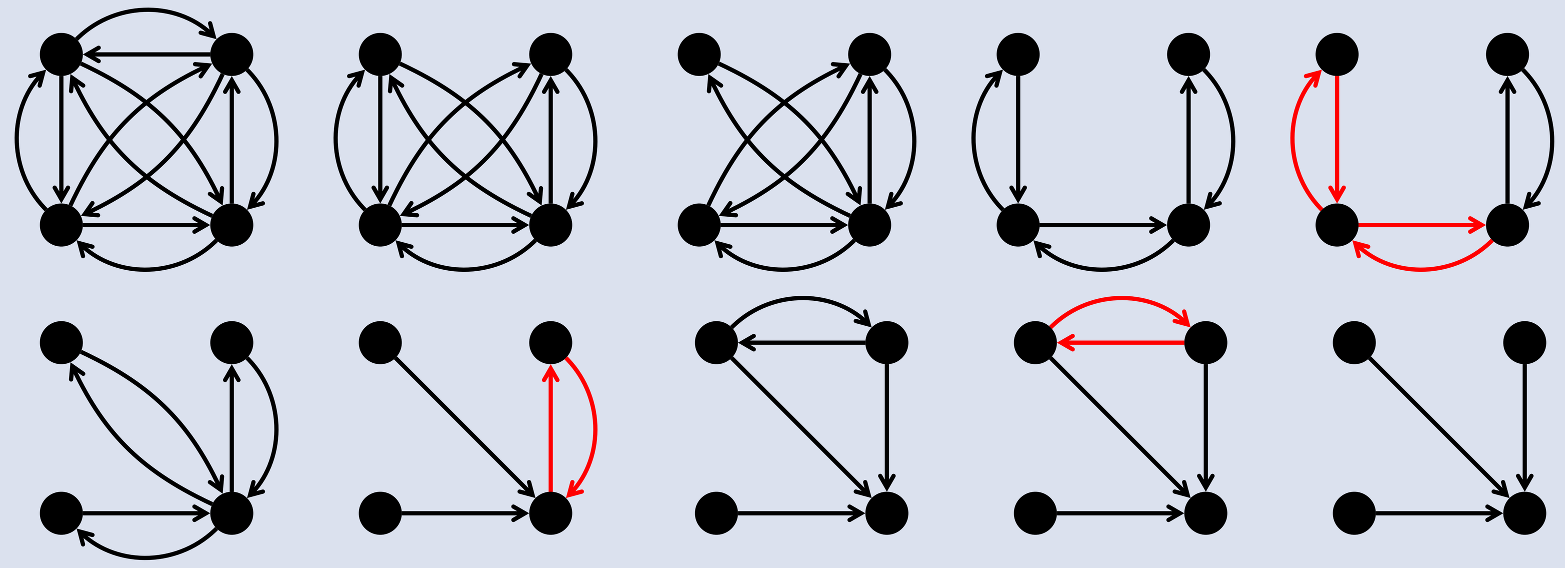


Figure: Commonly found motifs of size 4 with 1 color (all calls) or 2 colors (broadcast/point-to-point calls).

Machine Learning

Finally, we train several statistical models on our communication patterns and evaluate them using a maximum likelihood framework on out-of-band samples. These models include Hidden Markov Models, Naïve Bayes, Tree Augmented Naïve Bayes, and Random Forests.

The former three are types of Bayesian networks that calculate the likelihood of samples using a graph-based representation of dependencies between communication features. In contrast, random forests are ensembles of decision trees whose combined predictions are more accurate than individual trees.

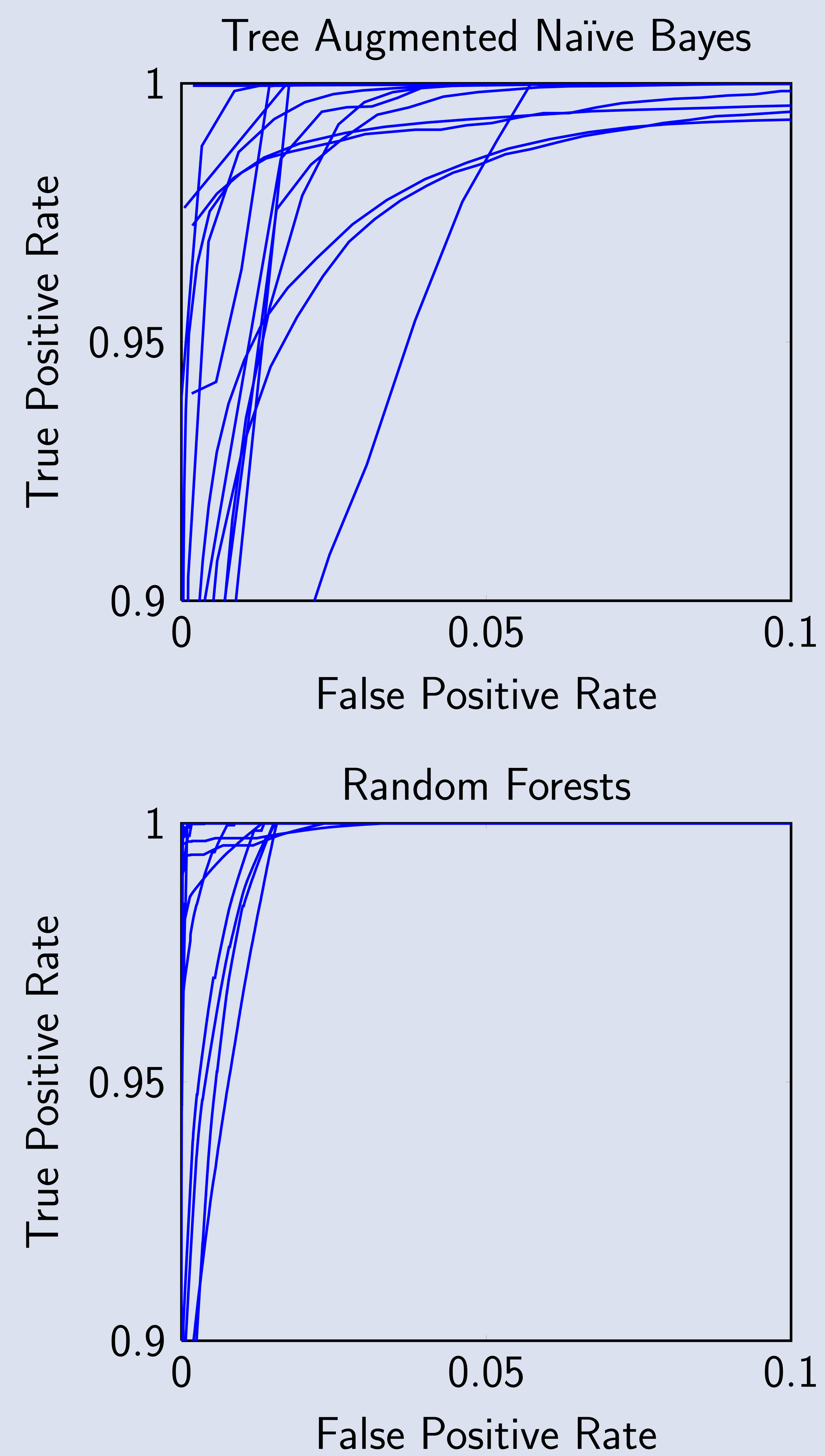


Figure: ROC visualization of 14 classifiers using Tree Augmented Naïve Bayes and Random Forests. The point $(0, 1)$ corresponds to ideal classification.

These models achieve nearly ideal classification for 14 different parallel programs. Each classifier was evaluated with 10-fold cross-validation to ensure reported accuracy is not a result of over-fitting. These models outperform call- and motif-based classifiers by 10-15%.

This research was supported in part by the Director, Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy, under contract number DE-AC02-05CH11231, and also by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001 under the auspices of the Institute for Information Infrastructure Protection (I3P) research program. The I3P is managed by Dartmouth College. The views and conclusions contained in this document are those of the authors and not necessarily those of its sponsors.