Enhancing tree-based algorithms performance

Efthyvoulos Drousiotis
EPSRC Centre for Doctoral Training in Distributed Algorithms, University of Liverpool, Liverpool, UK

Novel Decision Forest Building Techniques by Utilising Correlation Coefficient Methods

(Efthyvoulos Drousiotis, Lei Shi, Paul G. Spirakis, and Simon Maskell)

(Efthyvoulos Drousiotis, Lei Shi, Paul G. Spirakis, and Simon Maskell)

- Maximal Information Coefficient (MIC)

A powerful approach to measuring the correlation between two features. MIC can deal with the correlation analysis of linear, nonlinear, and potential non-functional relationships in large datasets

- Pearson's Correlation Coefficient Forest (PCCF)

In statistics, Pearson's correlation coefficient is used to measure the statistical relationship or correlation among variables. It is based on the covariance matrix of the data to determine the strength of the connection between two vectors.

Dataset	PCCF	MICF	BG	RS	RF	RFW p=1	RFW p=2
AB	21.3 (7.0)	27.5 (1.0)	25.1 (2.5)	25.0 (4.5)	25.0 (4.5)	25.1 (2.5)	23.7 (6.0)
AR	81.2 (7.0)	82.8 (4.0)	81.6 (6.0)	83.7 (1.0)	83.0 (2.5)	82.6 (5.0)	83.0 (2.5)
BS	84.2 (1.5)	84.2 (1.5)	77.5 (6.0)	72.2 (7.0)	80.5 (5.0)	81.1 (4.0)	82.5 (3.0)
DER	96.0 (2.0)	96.7 (1.0)	88.5 (4.0)	89.0 (3.0)	87.0 (7.0)	87.5 (5.0)	87.3 (6.0)
GI	76.7 (2.0)	77.2 (1.0)	74.1 (3.5)	73.2 (5.5)	74.1(3.5)	73.2 (5.5)	72.2 (7.0)
ION	93.7 (3.0)	94.5 (1.0)	92.6 (7.0)	93.4 (5.0)	93.7 (3.0)	93.7 (3.0)	92.9 (6.0)
LD	72.1 (2.0)	73.6 (1.0)	68.7 (6.0)	69.8 (5.0)	71.5 (3.0)	71.0 (4.0)	67.3 (7.0)
LC	76.6 (2.0)	84.4 (1.0)	63.9 (7.0)	68.9 (4.5)	68.9 (4.5)	68.9 (4.5)	68.9 (4.5)
PID	76.2 (2.5)	77.9 (1.0)	75.6 (6.0)	76.2 (2.5)	75.9 (4.0)	75.6 (6.0)	75.6 (6.0)
SCD	83.7 (3.0)	84.3 (1.5)	80.0 (6.5)	82.9 (4.5)	80.0 (6.5)	84.3 (1.5)	82.9 (4.5)
TAE	60.6 (2.0)	62.4 (1.0)	53.6 (7.0)	59.5 (3.0)	56.3 (4.0)	54.3 (5.0)	54.2 (6.0)
YST	60.3 (2.0)	60.9 (1.0)	60.5 (3.0)	58.6 (5.0)	59.5 (4.0)	57.9 (6.0)	48.9 (7.0)
Average	73.9 (3.0)	**75.7 (1.3)**	70.1 (5.4)	71.0 (4.2)	71.4 (4.3)	71.3 (4.3)	69.9 (5.5)

Single MCMC Chain Parallelisation on Decision Trees

(Efthyvoulos Drousiotis, Paul G. Spirakis)

Given a Decision's Tree MCMC chain with N iterations, we propose a method that utilises C number of cores aiming to enhance the running time of a single chain by at least an order of magnitude. At each iteration, a new sample χ′ is drawn from the proposal distribution. Our method requires sampling from C number of cores, S(C = S) number of samples in parallel. We then accept the sample with the greatest a(Ti, T ′) and repeat the same method until the Markov Chain converges to a stationary distribution. In our method, we check the Markov chain convergence when the F1-score fluctuates less than ±3% for at least 100 iterations. Once the Chain has converged, we proceed to the second phase of our method. We now keep producing samples using C cores, but we can now collect more than one sample which satisfies a(Ti, T ′) >= u. (u is a random uniform number[0, 1]) From this point, we will propose new samples from the sample with the greatest a(Ti, T ′) until we are happy with the number of samples collected. Using this method, we can collect the same number of samples and explore the feature space as effectively as the serial implementation, but 18 times faster.

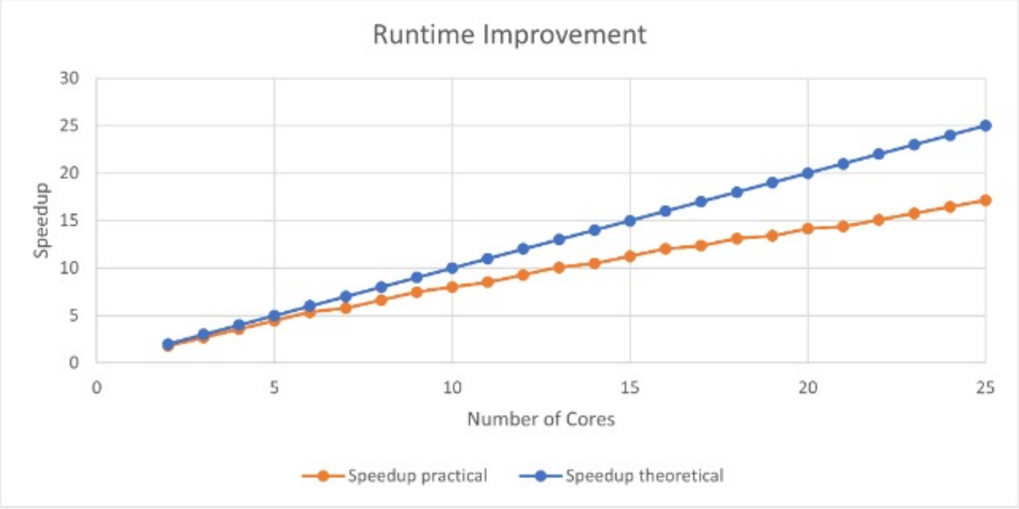

Fig. 3. Speedup achieved by utilising different number of cores

Single MCMC Chain Parallelisation on Decision Trees

(Efthyvoulos Drousiotis, Panagiotis Pentaliotis, Lei Shi, Alexandra Cristea)

- a wide-scale analysis, showing, firstly, that discrete-feature methods outperform sequential time-series methods, on both discrete and sequential datasets.
- Secondly, we show that this result is further consistent, when performing model hyperparameter optimisations and optimal feature engineering.

Supporting the more generic approach to converting the dataset, whenever possible, into the appropriate formats (in our case, time-series into discrete), which helps a different kind of predictive model than the default applied in previous studies, achieving faster training, testing, and predicting, as well as higher predictive accuracy and in general better performance compared to using multi-layer neural networks.

Models Used: 1) LSTM
2)Decision Tree
3)Random Forest
4)BART

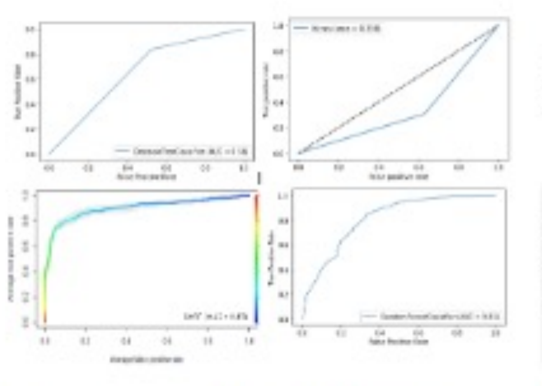

Fig. 1. Discrete data: No Pre-Processing, No Hyperparameter Optimisation

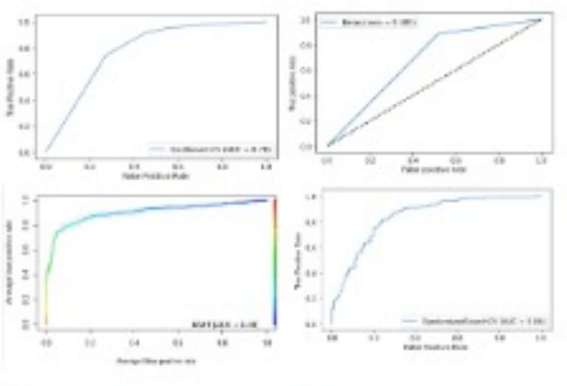

Fig. 2. Discrete data: Hyperparameter Optimisation, Feature Engineering

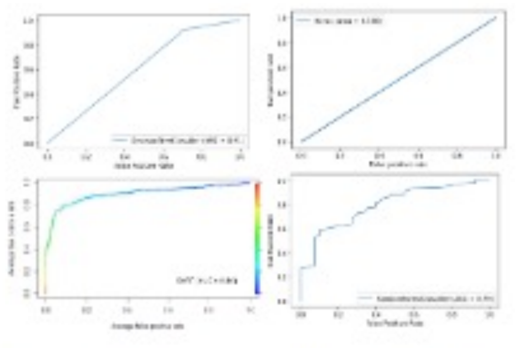

Fig. 3. Continuous data: No Pre-Processing, No Hyperpar. Optimisation

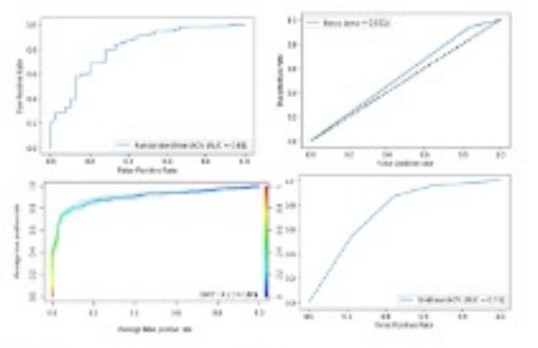

Fig. 4. Continuous data: Hyperparameter Optimisation, Features Engineering

(**For all ROC Curves above:** Upper left - Decision Tree, Upper right - LSTM, Bottom Left - BART, Bottom Right - Random Forest)