A Significantly Faster Elastic-Ensemble for Time-Series Classification

George Oastler and Jason Lines

University of East Anglia, Norwich, United Kingdom
{g.oastler, j.lines}@uea.ac.uk

Abstract. The Elastic-Ensemble [7] has one of the longest build times of all constituents of the current state of the art algorithm for time series classification: the Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTE) [8]. We investigate two simple and intuitive techniques to reduce the time spent training the Elastic Ensemble to consequently reduce HIVE-COTE train time. Our techniques reduce the effort involved in tuning parameters of each constituent nearest-neighbour classifier of the Elastic Ensemble. Firstly, we decrease the parameter space of each constituent to reduce tuning effort. Secondly, we limit the number of training series in each nearest neighbour classifier to reduce parameter option evaluation times during tuning. Experimentation over 10-folds of the UEA/UCR time-series classification problems show both techniques and give much faster build times and, crucially, the combination of both techniques give even greater speedup, all without significant loss in accuracy.

Keywords: time series · classification · ensembles · distance measures

1 Introduction

The current state of the art classifier in time series classification (TSC) is the Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTE) [8]. The Elastic-Ensemble (EE) [7] is one of five constituent classifiers in the HIVE-COTE meta-ensemble and key in uncovering discriminatory features in the time-domain. The discovery of these features leverages 11 nearest-neighbour classifiers (NN) each coupled with an *elastic* distance-measure. Consequently, EE requires a large amount of time to train and forms a bottle-neck in training HIVE-COTE. Ten of the constituent classifiers in EE use distance measures with $O(m^2)$ run-time complexity (where m is the length of the time-series). Eight of these each require parameter tuning using leave-one-out-cross-validation (LOOCV) over 100 parameter options. Therefore the tuning complexity of EE becomes $O(n^2m^2)$, an often impractically expensive procedure.

Distance-measures have been studied for a long time in TSC research and various distance-measure specific speed-ups have been conceived, such as utilising lower-bounds. However, further speed-ups can be made to the NN and parameter tuning aspects of EE such as:

1. using less parameter options when tuning constituent classifiers. We hypothesise many parameter options perform similarly, therefore a reduced parameter pool still contains a suitable parameter option.
2. using less train cases when estimating the accuracy of different parameter options during tuning. We hypothesise that training a NN on a subset of the train data will perform sufficiently well to evaluate a parameter option and maintain the relative ranks of parameter options during tuning.

We conducted experiments on various configuration of EE to investigate the effectiveness of these techniques. Our experiments use the UEA/UCR TSC problems [2] resampled 10 times at the original train/test distribution. First, we investigate the effectiveness of reduced parameter pools: 10%, 50% and 100% of the original parameter pool, chosen arbitrarily. We use the full training set to evaluate each parameter option. Second, we investigate the effectiveness of reduced training sets for parameter tuning: 10%, 50% and 100% of the original train set size, again chosen arbitrarily. We use the full parameter pool during tuning. Finally, we combine both techniques to investigate both reducing the parameter space and reducing the train set for each parameter option evaluation.

Our results demonstrate that either technique results in substantial reduction of training time without significant loss in test accuracy. Crucially, the subsequent combination of both techniques show further reduction of train time whilst still maintaining no significant loss in test accuracy. We conclude that limiting parameter pool size and train set size during tuning can speed-up EE by nearly two orders of magnitude without any significant loss in test accuracy.

2 Background and Related Work

A time-series, $T < x_1, x_2, x_3, ..., x_l >$, is an ordered sequence of l values with $x_i \in \mathbb{R}$. TSC is the task of prediction a class given a previously unseen time-series for which the class is unknown. A TSC classifier is therefore a function which is learned from the labelled time-series, the train set, to take an input of an unlabelled time-series, a test case, and output the predicted label.

TSC is an active area of research where many diverse algorithms have been proposed. These include, but are not limited to, histogram-based approaches that discriminate cases based on the frequency of reoccurring patterns [10,11]; shapelet algorithms that differentiate class membership through the presence of discriminatory, phase-independent subsequences [13,4]; and forest ensembles built on data transformed into different representations [8,3]. Arguably, most TSC research effort over the last decade has been focused on developing *elastic* distance measures to couple with simple 1-nearest neighbour (1-NN) classifiers. Such *elastic* measures are able to mitigate misalignments and phase-shift within time-series. The most common approach is to use Dynamic Time Warping (DTW) with a warping window set through cross-validation and a 1-NN classifier. Related variants of DTW have also been proposed, such as applying a soft boundaries to warping windows through weighting penalties [5], and warping directly on first-order derivatives [6]. Alternatives exist that are derived from

the edit-distance [1], and further hybrid measures have characteristics of both DTW and edit-distance [9,12].

2.1 The Elastic Ensemble

The performance of alternative *elastic* distance measures with 1-NN classifiers were compared in [7] to determine whether one approach outperformed all others. The measures included were: Dynamic Time Warping (DTW), Derivative DTW (DDTW), weighted variants of both DTW and DDTW (WDTW, WDDTW), Edit Distance with Real Penalty (ERP), Longest Common Subsequence (LCSS), Time Warp Edit (TWE) and Move-Split-Merge (MSM). All measures were coupled with 1-NN classifiers and the eight distance-measures with parameters were tuned over 100 parameter options each, respectively. Euclidean distance, full-window DTW and full-window DDTW were used as baselines and all eight subsequent 1-NN classifiers were compared over 85 TSC datasets [2]. It was found that no single measure significantly outperformed all others in test accuracy. However, the diversity in performance of each distance-measure inspired the EE, an ensemble of the 11 1-NN classifiers described above. In training, each constituent is evaluated using a LOOCV to obtain an estimate of train set accuracy and the optimal distance-measure parameter option if required. In testing, each constituent is given a vote weighted by its training accuracy estimate and the ensemble predicts the class with the greatest weighted vote.

3 Proposed Enhancements

3.1 Reduced parameter pool size

Eight of the distance measures in EE have a corresponding pool of 100 parameters. We hypothesise that there is a large amount of redundancy due to the similarity in parameter option performance, therefore the pool of parameters can be reduced whilst still yielding a suitable parameter choice during tuning. In our experiments we arbitrarily use 10%, 50% and 100% of the full parameter pool, sampled randomly.

3.2 Reduced neighbourhood size

LOOCV is used to evaluate each parameter option during tuning of the eight distance measures which required parameters. We hypothesise that parameter options can be effectively evaluated in a NN using substantially less train cases, hence reducing the impact of the expensive LOOCV procedure for parameter tuning. Less training cases reduces the neighbourhood of potential nearest neighbours in the NN, decreasing test time and speeding-up LOOCV. We believe a sufficiently large neighbourhood should evaluate a parameter option accurately enough to maintain ranks of parameter options during tuning. In our experiments we arbitrarily use 10%, 50% and 100% of the training set, sampled randomly.

3.3 Combined Strategies

A subsequent technique is to reduce both the parameter pool size 3.1 and neighbourhood 3.2 size. The effectiveness of this technique is dependent upon the success of the previous techniques. The two techniques likely impact each other, as introducing less-accurate parameter evaluation through a limited neighbourhood size may not find the optimal parameter option in a reduced parameter option pool. We designed a subsequent experiment to assess all combinations of these techniques over the 10%, 50% and 100% limits of parameter pool size and neighbourhood size respectively.

4 Experimental Design

We ran experiments to assess the impact of techniques from Section 3. Our experiments use the UEA/UCR TSC problems over 10 resamples at the original train/test ratio. Only 48 of the smallest datasets were investigated due to infeasible run-time of full EE, demonstrating the importance of speeding up EE for realistic usability. These results are indicative of the performance of the training strategies however, and further results will be added in due course when available to confirm the findings over the complete repository. All source code can be downloaded from the provided link[1][2].

5 Results

The results are organised into three separate experiments and findings: parameter pool size reduction, neighbourhood size reduction, and combining both reduction techniques. For each experiment we report the accuracies and train-times to assess the impact of each technique upon the performance of EE.

The critical difference (CD) diagrams used throughout these experiments visually demonstrate the results of comparing all classifiers using pairwise Wilcoxon signed rank tests, where *cliques* are formed using the Holm correction to represent classifiers where there is no significant difference between them.

5.1 Parameter pool size reduction

The results shown in the CD diagram of Figure 1 indicate that using 10% of possible parameter options during tuning of each EE constituent is not significantly different to original EE which uses all possible parameter options.

This demonstrates that EE can be trained much faster with no significant loss in test accuracy. It is worth noting there is a significant difference in 10% and 50% parameter pool sizes. This indicates the random sampling of 10% of the

[1] https://github.com/TonyBagnall/uea-tsc/commit/07408d166072e8fd3057cb1fcbfd913e603094e3

[2] https://github.com/alan-turing-institute/sktime

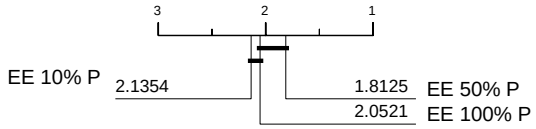
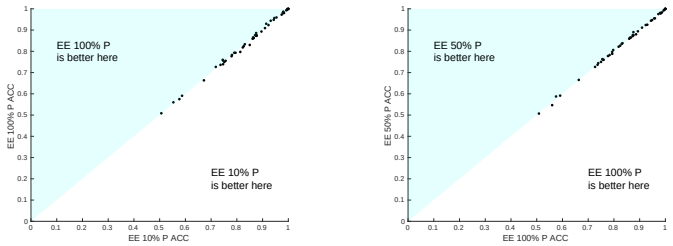


Fig. 1: A CD diagram to compare the test accuracy of EE using reduced parameter pool sizes during tuning of each constituent NN of EE. For clarity, "10% P" uses only 10% of the available parameter pool during tuning for each constituent classifier.



(a) Scatter plot comparing test accuracy of full EE against EE with 10% parameter pool.

(b) Scatter plot comparing test accuracy of full EE against EE with 50% parameter pool.

Fig. 2

parameter pool may be too few and further investigation is required. We provide scatter plots in Figure 2 comparing the two reduced parameter pool sizes against full EE to demonstrate no significant difference in test accuracy.

5.2 Neighbourhood size reduction

The results of reducing the neighbourhood size during tuning of parameter options are summarised in the CD diagram in Figure 3 and scatter plots in Figure 4.

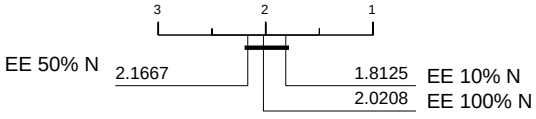
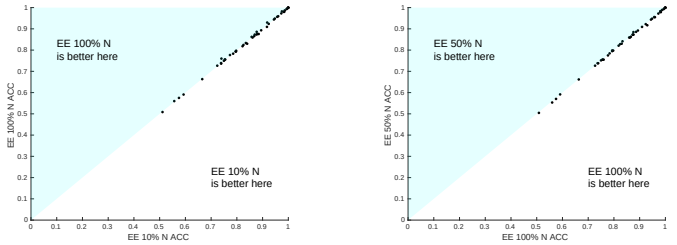


Fig. 3: A CD diagram to compare the test accuracy of EE with reduced neighbourhood sizes for evaluating parameter options during tuning of each constituent NN. For clarity, "10% N" uses 10% of the training data during tuning to evaluate a parameter option.

The results demonstrate that there is no significant loss in test accuracy when using 10% or 50% of training data during tuning of parameter options. This confirms our hypothesis that parameter options can be sufficiently evaluated and ranked using much less training data. Therefore, a substantial amount of time



(a) Scatter plot comparing test accuracy of full EE against EE with 10% neighbourhood size.

(b) Scatter plot comparing test accuracy of full EE against EE with 50% neighbourhood size.

Fig. 4

can be saved while training EE by using a smaller neighbourhood. We reinforce the equivalence in test accuracy in the scatter plots in Figure 4.

5.3 Combined techniques: reduced parameter pool size and reduced neighbourhood size

The results presented in Sections 5.2 and 5.1 can be combined to further speed-up the training of EE. These results show there is no significant difference between full EE against EE with 10% parameter pool size, and no significant difference between full EE against EE with 10% neighbourhood size. These techniques can be combined to investigate further speed-up, again arbitrarily using the values of 10% and 50% to produce four combinations of each (10%/10%, 10%/50%, 50%/10%, and 50%/50%) alongside full EE with 100% neighbourhood size and 100% parameter pool size. These results are summarised in the critical difference diagram in Figure 5 and scatter plot in Figure 6.

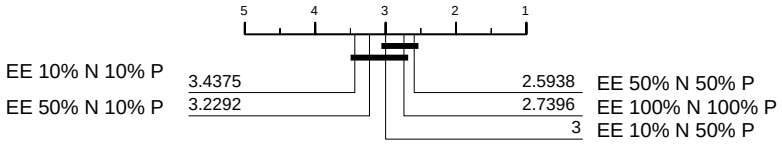


Fig. 5: A CD diagram to compare the test accuracy performance of EE using various neighbourhood sizes (N) and parameter pool sizes (P). For clarity, 50% N and 10% P corresponds to tuning each constituent NN of EE using 10% of the full parameter pool and evaluates each parameter option 50% of the training data.

The results in Figure 5 confirm that there is no significant difference in the test accuracies of full EE and EE trained using only 10% of parameter options and 10% of training data during tuning (EE-10%). This does not lower the run-

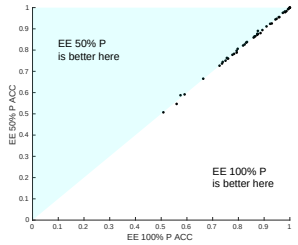


Fig. 6: Scatter plot comparing test accuracy of full EE against recommended EE configuration with 10% neighbourhood size and 50% parameter pool size.

Table 1: Table of accuracies and timings over UEA/UCR TSC datasets comparing EE with different configurations of neighbours and parameters. N corresponds to percentage of neighbours, P corresponds to the percentage of parameters.

Dataset	100% N 100% P	50% N 50% P	10% N 50% P	50% N 10% P	10% N 10% P
Accuracy (%)	0.8478	0.8491	0.8477	0.8466	0.8465
Train time (minutes)	37.1200	17.8101	2.4314	15.1747	0.9948
Train time (% of full EE)	100.0000	47.9800	6.5500	40.8800	2.6800

time complexity of EE but does decrease the train time to approximately 3% of full EE as outlined in Table 1.

Note that the test accuracy of EE-10%, whilst not significantly different to full EE, is significantly worse than the best performing variants which use 50% of parameter options. Table 1 indicates reducing neighbourhood size provides better speed-up than reducing parameter pool size. Therefore, reducing parameter pool size beyond 50% whilst also reducing neighbourhood size significantly decreases test accuracy. Practitioners are advised to use 50% parameter pool size and 10% neighbourhood size as a sufficient compromise to reduce train time whilst preserving test accuracy. The recommended EE with 50% parameter pool size and 10% neighbourhood size was ranked 3rd overall in Figure 5 and was not significantly outperformed by any other technique. Furthermore, the timing results in Figure 1 show that this configuration requires approximately only 6.6% of the time of full EE - nearly two orders of magnitude faster than the original EE. We demonstrate the equivalence in test accuracy between the recommended configuration and the full EE in Figure 6.

6 Conclusions and future work

In this work we have investigated two techniques for reducing the training time required to run EE. First, we proposed a technique to reduce the distance measure parameter pool size (using random sampling) for each constituent NN classifiers of EE during tuning. Second, we proposed a technique to use less neighbours

(via random sampling) in the NN constituent classifiers to evaluate a parameter option whilst tuning. We hypothesised that both could lead to substantial speed-ups in train times without significant loss in accuracy as a suitable parameter option is still found. We validated these claims through two independent experiments looking at either technique and conclude that using either 10% neighbourhood size or 10% parameter pool size does not significantly reduce test accuracy.

Inspired by these findings, we combined both techniques to build EE with a reduced parameter pool size and neighbourhood size. We found that EE could be sped-up to approximately 3% of the original train-time of full EE at best. We also conclude that using the recommended, and crucially not significantly worse, configuration of 10% neighbourhood size and 50% parameter pool size takes approximately only 6.6% train-time versus full EE.

References

1. Chen, L., Ng, R.: On the marriage of Lp-norms and edit distance. In: Proc. 30th International Conference on Very Large Databases (VLDB) (2004)
2. Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G.: The UEA-UCR time series classification archive. `http://www.cs.ucr.edu/~eamonn/time_series_data/` (2015)
3. Deng, H., Runger, G., Tuv, E., Vladimir, M.: A time series forest for classification and feature extraction. Information Sciences **239**, 142–153 (2013)
4. Hills, J., Lines, J., Baranauskas, E., Mapp, J., Bagnall, A.: Classification of time series by shapelet transformation. Data Mining and Knowledge Discovery **28**(4), 851–881 (2014)
5. Jeong, Y., Jeong, M., Omitaomu, O.: Weighted dynamic time warping for time series classification. Pattern Recognition **44**, 2231–2240 (2011)
6. Keogh, E., Pazzani, M.: Derivative dynamic time warping. In: Proc. 1st SIAM International Conference on Data Mining (SDM) (2001)
7. Lines, J., Bagnall, A.: Time series classification with ensembles of elastic distance measures. Data Mining and Knowledge Discovery **29**, 565–592 (2015)
8. Lines, J., Taylor, S., Bagnall, A.: Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles. ACM Trans. Knowledge Discovery from Data **12**(5) (2018)
9. Marteau, P.: Time warp edit distance with stiffness adjustment for time series matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(2), 306–318 (2009)
10. Schäfer, P.: The BOSS is concerned with time series classification in the presence of noise. Data Mining and Knowledge Discovery **29**(6), 1505–1530 (2015)
11. Schäfer, P., Leser, U.: Fast and accurate time series classification with weasel. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 637–646. ACM (2017)
12. Stefan, A., Athitsos, V., Das, G.: The Move-Split-Merge metric for time series. IEEE Transactions on Knowledge and Data Engineering **25**(6), 1425–1438 (2013)
13. Ye, L., Keogh, E.: Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. Data Mining and Knowledge Discovery **22**(1-2), 149–182 (2011)