a summary of

Proximity Forest : An Effective and Scalable Distance-Based Classifier for Time Series

Lucas et. al, Data Min. & Knowl. Discov. 2019

●●●

by Ram Sai Ganesh

Paper linked here

More data than ever before

- Much research has gone into high-accuracy time-series classifiers
- UCR dataset – order of 1,000s or 10,000s of time series
  - More recent datasets – 370k (phoneme), 10^6 (satellite).
- Motivation for a performance-optimized algorithm
  - Lack of scalability in current SOTA
- Spoiler, Proximity Forests run really quick...
  - Linear w.r.t. training set size! Upto 100,000 times faster than EE & COTE on occasion.

Background

- Distance methods:
  - Distances – extensively studied. Tend to require re-aligning the time series to remove non-linear distortions.. Includes Dynamic Time Warping (improvements), LCSS and MSM.
  - NN Approaches – common to incorporate distance methods. Was the benchmark. Issues include param tuning (CV) & training cost. EE* is a recent SOTA 11-NN ensemble.
- Feature Learners:
  - Shapelets & Bag-of-Words – search for characteristic subseries. Current SOTAs – Shapelet Transform uses Euclidean distance attributes to transform shapelets for further classification. Doesn’t scale – O(n^2 . l^4); BOSS-VS converts TS->words using SFA; doesn’t scale beyond 10k TS.

Background

- Ensemble Approaches:
    - Combine multiple classifiers. COTE is a 35-ensemble 4 domain classifier, the current UCR dataset SOTA. However, time complexity is bound by the component Shapelet Transform – O(n^2 . l^3)
- Decision Tree Approaches:
    - TSF – split TS into intervals, create summary, use a Rand Forest – split TS at each node based on random features.. Virtue is runtime – O(n . log(n) . l . k), k trees.
    - Generalized Random Shapelet Forest – threshold based on distance measure of a random shapelet compared to other series, split until node is pure. Computationally expensive to consider more candidate samples, and limited by shapelet performance.

Proximity Forests (PF)

Consider ‘n’ labelled time series of length ‘l’ each, labels are 1 to c.

- A Proximity forest is an ensemble of ‘k’ Proximity Trees (PTs).

While a regular decision tree applies a test based on an attribute, each branch of a PT holds an exemplar; an object follows the branch to which it is closest. A Tree is either a leaf or an internal node.

- A PT creates at each node a class for each class of data it receives.

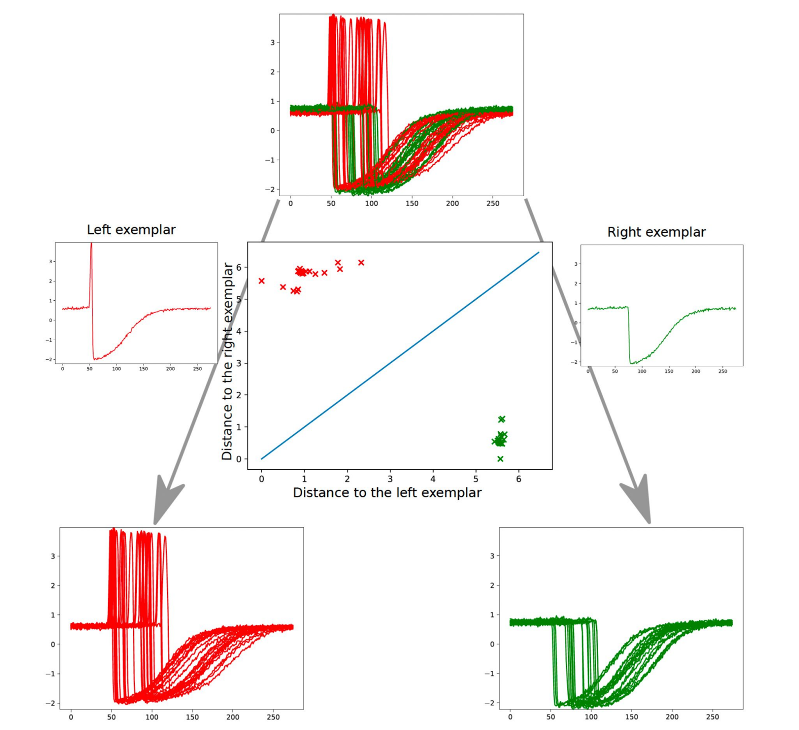
At each node, a group of ‘r’ candidate splits are evaluated.

Proximity Forests (PF)

- We select an exemplar from each class and pass the data down to the branches.
- Once all ‘r’ candidate splits have been created, we select the candidate that maximize the difference between the Gini impurity of the parent node and the weighted sum of the Gini Impurities of the child nodes.
- Continue to select candidates and spilt until pool is exhausted.

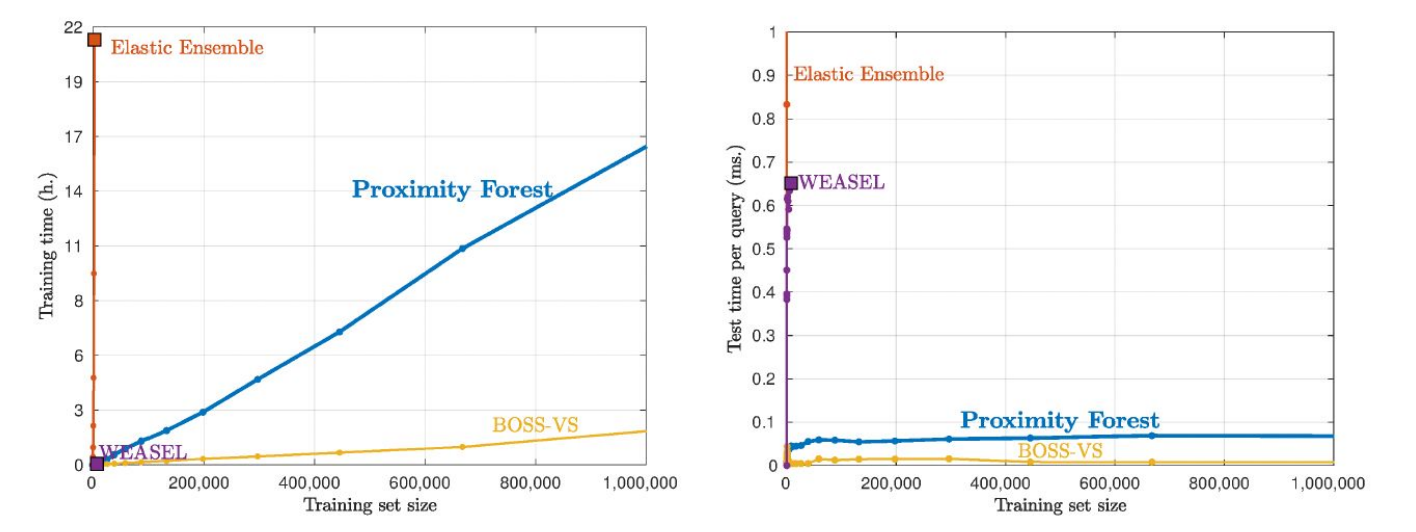
Notes

- The parameterized distance measure is chosen from the same 11 used by EE.
- Parameters for the distance measures were chosen to closely resemble EE for comparison.



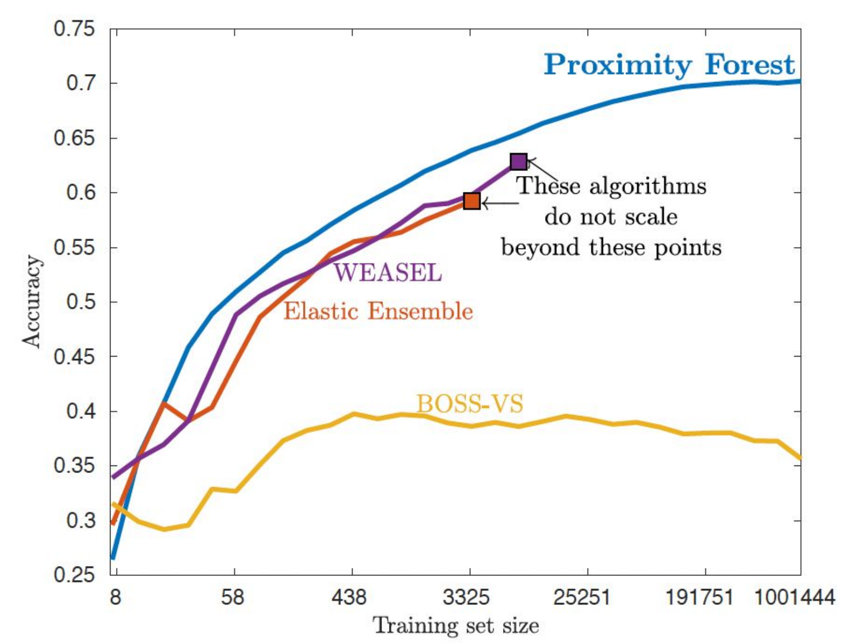
Results

- In practice, we expect an average training complexity of O(k n log(n) r c l^2).

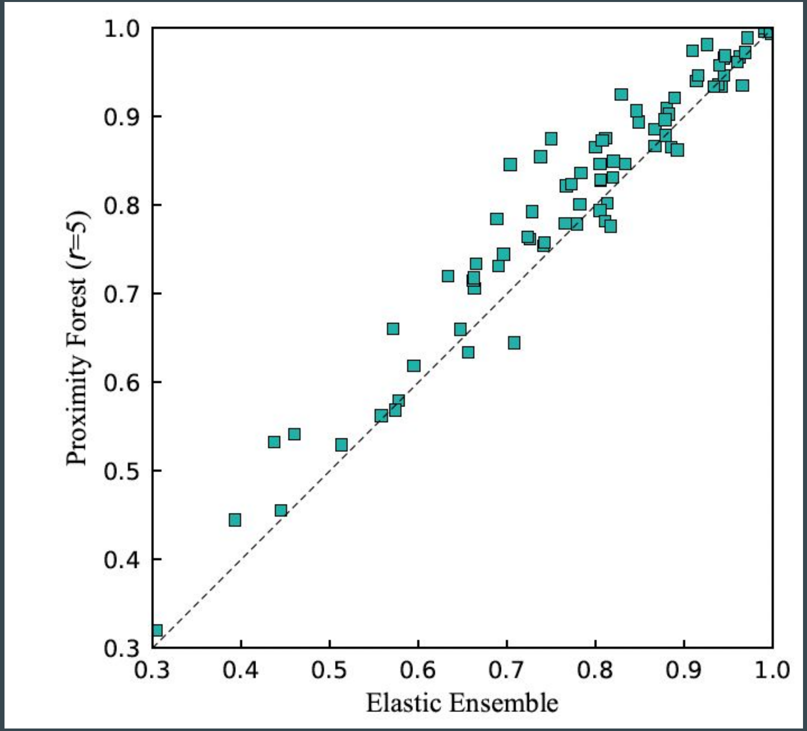


Results

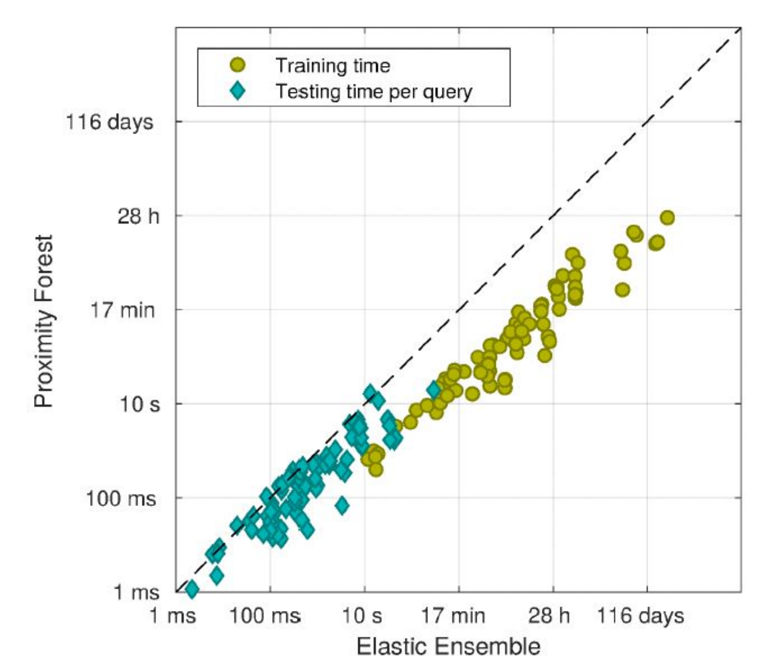
- PFs scale logarithmically with training set size while EE must scan the full database many times.
- This performance reflects on the UCR dataset as well, outclassing other less efficient algorithms.



Results



Proximity Forest (r=5)
Elastic Ensemble
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0



Training time
Testing time per query
Proximity Forest
Elastic Ensemble
1 ms
100 ms
10 s
17 min
28 h
116 days

Citations

i. Lucas, B., Shifaz, A., Pelletier, C. *et al.* Proximity Forest: an effective and scalable distance-based classifier for time series. *Data Min Knowl Disc* 33, 607–635 (2019). https://doi.org/10.1007/s10618-019-00617-3

Questions?