A Time Series Forest for Classification and Feature Extraction

Reviewed by Stephen Lauer

April 3, 2017

Preface: a misclassification

- This paper was about classifying time series, not about making time-series forecasts
  - Whoops!
- However, this did give me some ideas for extensions of random forests into time-series forecasts

Introduction

- N time series each of length M, which we want to put into C classes
  - e.g. heart rate time series, some of which have arrhythmia
- Can create a "feature" (covariate) across any two time intervals
  - e.g. the mean between time 10 and time 30
- If there are K feature types, there are KM^2 possible features
  - need a very large forest to fully explore interval feature-space
- Many time intervals are highly correlated, need to find most important

Decision Tree Sampling

- Typically in RF, if there are p covariates to choose from, each decision tree in the forest will randomly sample $\sqrt{p}$ of them
- In TSF, $\sqrt{M}$ intervals and $\sqrt{M}$ starting points are chosen (for each interval length?), then p covariates are made for each
  - For a total of Mp covariates
- So a timeseries with $M = 1200$ and $p = 3$ would have 3600 covariates of a possible 4.3 million
  - This still seems like a lot, right?

Splitting RF decision trees

- We want each leaf (or node) of a tree to end up with the same class
- Entropy gain is a commonly used splitting criterion in RF

$$\text{Entropy} = -\sum_{c=1}^{C} \gamma_c \log \gamma_c$$

- If all nodes are homogenous (each node has one class), Entropy=0
- If all nodes are perfectly heterogenous (even mixing at node level), Entropy=1
- Entropy gain is $\Delta\text{Entropy} = \text{Entropy}_{\text{parent}} - \text{Entropy}_{\text{child}}$

Splitting RF decision trees (Example)

- Parent has one node with all cases 5 class A, 9 class B

$$
\begin{aligned}
\text{Entropy}_{\text{parent}} &= \mathit{Entropy}(5, 9) \\
&= -\frac{5}{14}\log\frac{5}{14} - \frac{9}{14}\log\frac{9}{14} \\
&= 0.904
\end{aligned}
$$

Splitting RF decision trees (Example)

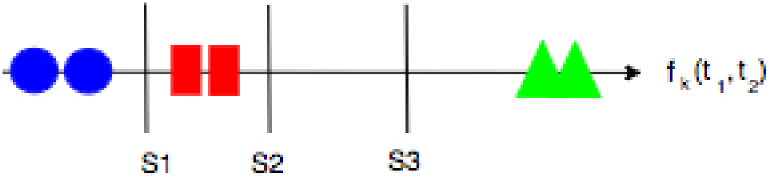
- Child splits up into 2 nodes, with 4 class A, 2 class B in one and 1 class A, 7 class B in the other:

$$\begin{aligned}
\text{Entropy}_{\text{child}} &= \frac{6}{14} \mathit{Entropy}(4, 2) + \frac{8}{14} \mathit{Entropy}(1, 7) \\
&= \frac{3}{7}(0.637) + \frac{4}{7}(0.377) \\
&= 0.429
\end{aligned}$$

- Thus, $\Delta\text{Entropy} = 0.904 - 0.429 = 0.475$

An additional splitting criterion

These three splits all have the same entropy:



- The authors consider the “Margin” as an entropy tiebreaker
- The logic behind this is that this interval doesn’t actually differentiate red and blue very much, but it does show a split between green and red/blue
- There’s a good chance that a new red observation could be on the right side of the S2 split, so S3 is the “best” split

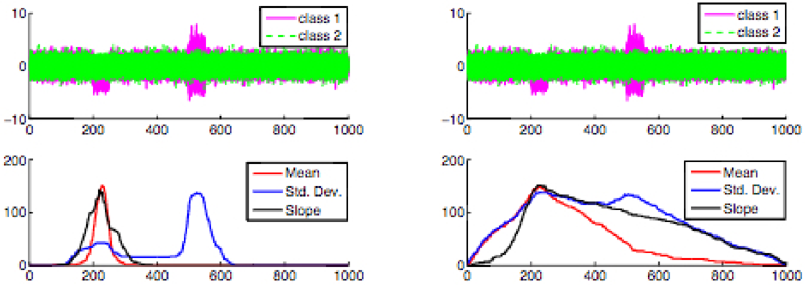
Temporal importance curves

Finally, we want to find the times when each feature made the most impact

$$\mathrm{Imp}_k(t) = \sum_{t_1 \le t \le t_2, \nu \in SN} \Delta\mathrm{Entropy}(f_k(t_1, t_2), \nu)$$

This measures the entropy gain at each time t in the interval t_1, t_2 for each node ν at any of the total SN nodes.

Temporal importance curves



Simulation results

- Tested TSF with 500 trees and functions mean, SD, and slope versus other time series classifiers on 45 sets of time series datasets
- Across all datasets, had the highest average rank of any classifier and was better than any other classifier one-on-one (had a higher "winning percentage" in head-to-head performances)
- Error rates for TSF fell quickly to 100 trees then leveled off
- Computation speed is linear with regard to both time series length and size of the training sample

How can we translate this to our work?

- The problem we usually deal with is just one time series, trying to forecast future points given a set of past points.
- One area where the techniques here could be useful would be in determining useful weather and incidence time frames
  - How much recent incidence is important?
  - Can we identify a “susceptible window” of some sort?
  - Can we find reliable weather windows?

How can we translate this to our work?

1. Use exact same methodology in the paper, by splitting up time series into many smaller time series
   - For a 50-year time series, we could make take 5-year run-ups to the following season (e.g. 1968-1972 to forecast 1973, 1969-1973 to forecast 1974, etc.)
   - We could classify years as high/low or high/medium/low
   - Or we could try to extend methodology into the continuous space (there are already continuous RF, shouldn't be hard)
   - Would the correlation between samples be a problem?
   - Could this work across different provinces?

How can we translate this to our work?

2. Use the sampling portion of the methodology to choose lagged time intervals instead of raw time intervals
   - Need to think about how to choose, since only $M - \ell$ data points have ℓ-lag covariates
   - Outcomes of decision trees could be 1-step to h-step horizons, with same issue
   - For Bangkok, we have $\sim$1250 biweeks, leave out last 10 years for testing (260 biweeks), predict up to 1 year forward (26 biweeks), and cap lags at 10 years back leaves us with $\sim$700 training points

Evaluation of TSF in other paper

- "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances" by Bagnall et al looked at 18 different time series classification (TSC) algorithms (of 100s in various papers), including TSF
- May or may not have implemented it correctly
- Found that using "Margin" as tiebreaker had a negative effect on prediction accuracy
- It placed in the second best group of TSC algorithms
- The temporal importance curve provides better feedback than some of the better competitors, but could dig deeper to see if one of these has interpretable results