Forecasting performance evaluation in time series instance spaces

Talk at BUAA

Yanfei Kang, Rob Hyndman and Kate Smith-Miles

Oct 25, 2017

Beihang University

Motivation

M3 data

International Journal of Forecasting 16 (2000) 451–476

international journal of forecasting

www.elsevier.com/locate/ijforecast

The M3-Competition: results, conclusions and implications

Spyros Makridakis, Michèle Hibon*

INSEAD, Boulevard de Constance, 77305 Fontainebleau, France

Abstract

This paper describes the M3-Competition, the latest of the M-Competitions. It explains the reasons for conducting the competition and summarizes its results and conclusions. In addition, the paper compares such results/conclusions with those of the previous two M-Competitions as well as with those of other major empirical studies. Finally, the implications of these results and conclusions are considered, their consequences for both the theory and practice of forecasting are explored and directions for future research are contemplated.

Keywords: Comparative methods — time series: univariate; Forecasting competitions; M-Competition; Forecasting methods, Forecasting accuracy

M3 data

- 3003 time series
- From demography, finance, business and economics
- Lengths between 14 and 126
- Either non-seasonal, monthly or quarterly
- Positive

Questions

- Do we favour forecasting methods that work well with specific types of data?
- How diverse and challenging are these time series?
- Are there particular features of some time series that make them particularly amenable to being forecast by one method compared to another?

What we do

- Visualize M3 data in feature space.
- Study the distribution of their features.
- Identify gaps in the instance space, and generate new time series with controllable features given a target location.
- Predict forecasting method performance in the instance space.

Time series features

Basic idea

Transform a given time series $\{x_1, x_2, \cdots, x_n\}$ to a feature vector $F = (F_1, F_2, \cdots, F_p)'$.

Why?

1. When the time series is very long, this is kind of dimension reduction.
2. It deals with time series with different lengths.
3. Focus on shapes.

Time series features

The six features used to characterize a time series.

1. Spectral entropy
2. Strength of trend
3. Strength of seasonality
4. Seasonal period
5. First order autocorrelation
6. Optimal Box-Cox transformation parameter

Spectral entropy F_1

We use an estimate of the Shannon entropy of the spectral density $f_x(\lambda)$ of a stationary process x_t:

$$F_1 = -\int_{-\pi}^{\pi} \hat{f}_x(\lambda) \log \hat{f}_x(\lambda) d\lambda,$$

where $\hat{f}_x(\lambda)$ is an estimate of the spectrum of the time series.

- Small $F_1 \Rightarrow$ more signal and more forecastable.
- Relative larger $F_1 \Rightarrow$ more uncertainty and harder to forecast.

Strength of trend F_2 and strength of seasonality F_3

STL decompostion

$$x_t = S_t + T_t + R_t.$$

The strength of trend can be measured by comparing the variances of R_t and $x_t - S_t$.

$$F_2 = 1 - \frac{\mathrm{var}(R_t)}{\mathrm{var}(x_t - S_t)}.$$

The strength of seasonality is defined as:

$$F_3 = 1 - \frac{\mathrm{var}(R_t)}{\mathrm{var}(x_t - T_t)}.$$

Seasonal period F_4

- $F_4 = 12$ for monthly data
- $F_4 = 4$ for quarterly data
- $F_4 = 1$ for nonseasonal data
- When the period is unknown, it could be estimated from the data using, for example, the `findfrequency()` function from the **forecast** package in **R**.

First order autocorrelation F_5

ACF is greatly affected by trend and seasonality, so we compute the autocorrelations in $\{R_t\}$:

$$F_5 = \text{Corr}(R_t, R_{t-1}).$$

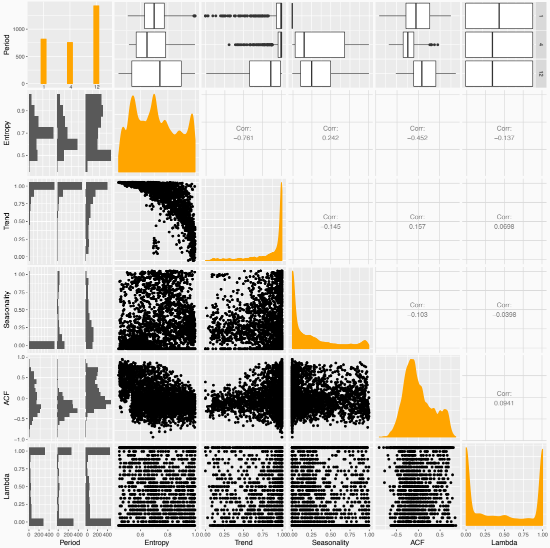
Optimal Box-Cox transformation parameter F_6

Box-Cox transformation

$$w_t = \begin{cases} \log(x_t), & \text{if } \lambda = 0, \\ (x_t^\lambda - 1)/\lambda, & \text{otherwise.} \end{cases}$$

- A good λ makes the variation of a series approximately constant across the whole series.
- We choose $\lambda \in (0, 1)$ to maximise the profile log likelihood of a linear model fitted to x_t.
- Measures the degree of change of variation in the data.

M3 time series features



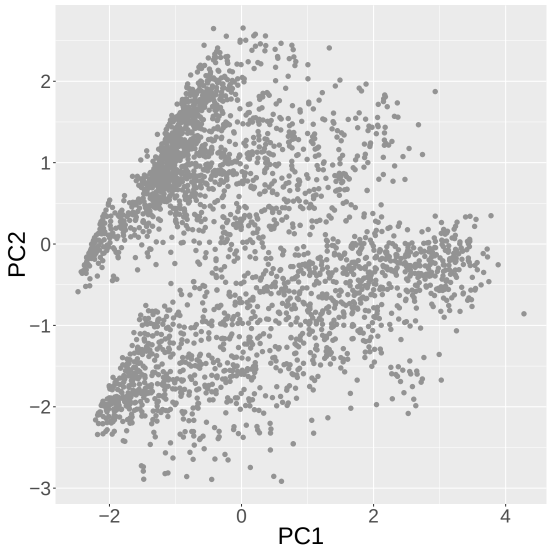
Visualisation

- PCA

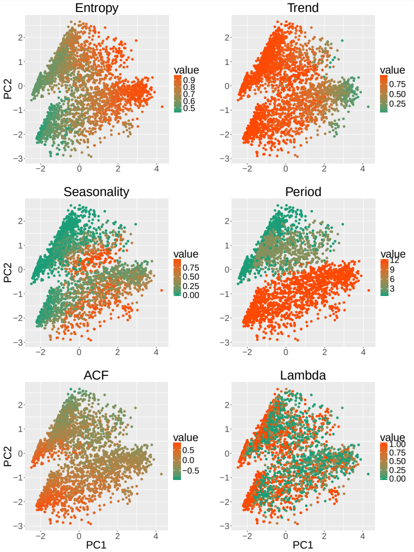
$$\begin{bmatrix} \mathrm{PC1} \\ \mathrm{PC2} \end{bmatrix} = \begin{bmatrix} 0.614 & -0.588 & 0.321 & 0.258 & -0.292 & -0.150 \\ 0.210 & 0.000 & -0.307 & -0.687 & -0.608 & -0.114 \end{bmatrix} \mathbf{F}$$

- PC1 increases with spectral entropy and decreases with trend and first order autocorrelation.
- PC2 negatively depicts period and seasonality.

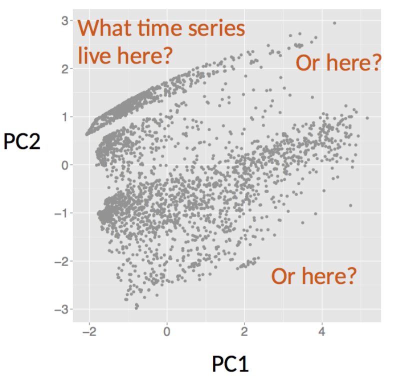
Feature space of M3



Feature distributions of M3



Questions



- Is it possible to fill and extend the whole space? or
- Can we generate a more diverse set of time series than M3?

New time series generation in feature space

New time series generation

- Once a target point is set, our goal is to evolve a new time series instance which is as close as possible to the target point.
- The process relies on a genetic algorithm (GA). Initial populations are improved until the final population is achieved with maximised fitness.

GA procedure

For each target point T_i, $i = 1, 2, \ldots, N_t$, we first generate an initial population of time series and iterate until the whole process meets some convergence criteria:

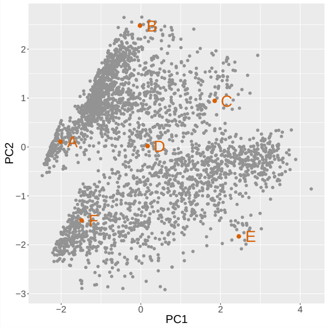
1. Calculate the feature vector for each time series $j \in \{1, 2, \ldots, N_p\}$ in the current population. Project it into 2-d: PC_j.
2. Calculate the fitness of each member in the current population:

$$\mathrm{Fitness}(j) = -\sqrt{(|\mathrm{PC}_j - T_i|^2)}.$$

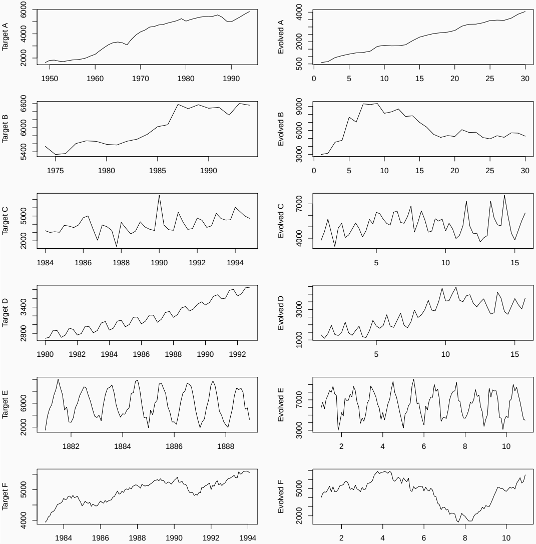
3. Evolve the next generation based on the fittest individuals.

From the final population, we select the instance closest to the target point.

Validation

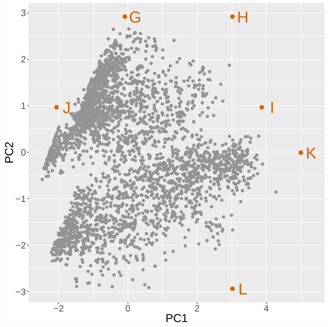


B
C
A
D
F
E
PC2
PC1
2
1
0
−1
−2
−3
−2
0
2
4

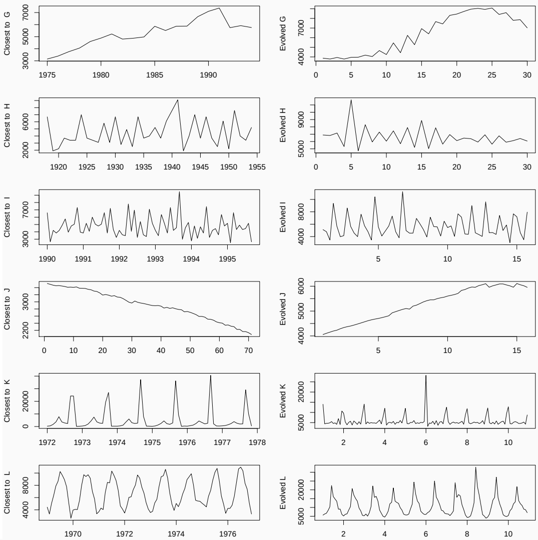


Target A
Evolved A
Target B
Evolved B
Target C
Evolved C
Target D
Evolved D
Target E
Evolved E
Target F
Evolved F

Validation



G
H
J
I
K
L
PC1
PC2

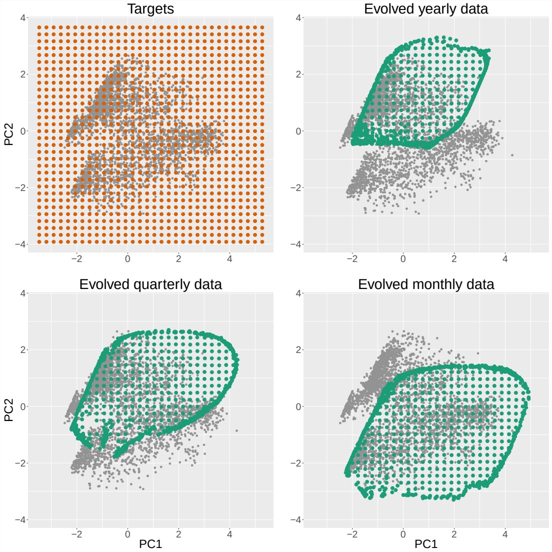


Closest to G
Evolved G
Closest to H
Evolved H
Closest to I
Evolved I
Closest to J
Evolved J
Closest to K
Evolved K
Closest to L
Evolved L

Target points

- 32 *32 grid with 1024 points, which are bounded within one unit wider than the upper and lower bounds of PC1 and PC2.
- Generate 1024 yearly, quarterly and monthly time series that are previously unknown and evolved by maximising the fitness function.

Results



Targets
Evolved yearly data
Evolved quarterly data
Evolved monthly data
PC2
PC1
4
2
0
−2
−4

Comparison of time series forecasting methods in the feature space

No-free-lunch

- There is never likely to be a single method that fits all situations.
- There is no time series forecasting method that will always perform best. Even for one particular time series, no one technique is consistently superior to others.

Time series forecasting methods

1. Naïve: using the most recent observation as the forecast.
2. Seasonal naïve: forecasts are equal to the most recent observation from the corresponding time of year.
3. The Theta method, which performed particularly well in the M3-Competition.
4. ETS: exponential smoothing state space modelling.
5. ARIMA: autoregressive integrated moving average models.
6. STL-AR: an AR model is fitted to the seasonally adjusted series, while the seasonal component is forecast using Seasonal naïve.

Minimum MASE of M3

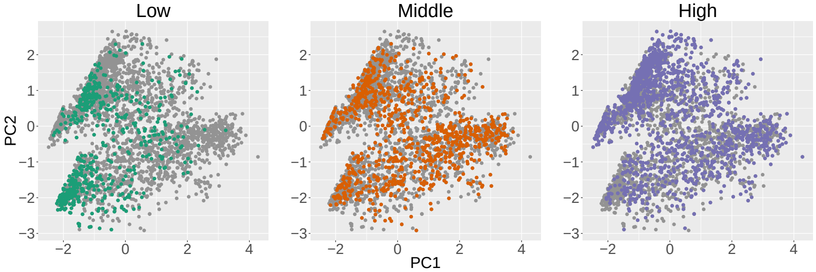


Figure 1: Locations of M3 series which achieve Low, Middle and High minimum MASE from all the six forecasting algorithms.

MASE of M3

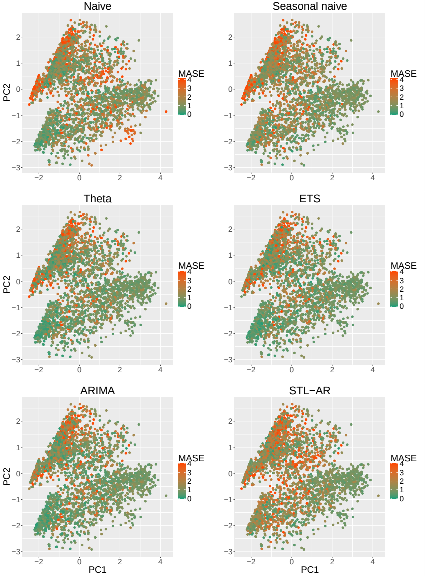
Forecasting method	Yearly	Quarterly	Monthly	Other	All
Naïve	3.17	1.46	1.17	3.09	1.79
Seasonal naïve	3.17	1.43	1.15	3.09	1.76
Theta	**2.77**	**1.11**	0.89	2.27	**1.43**
ETS	2.88	1.19	**0.86**	**1.82**	**1.43**
ARIMA	2.96	1.19	0.88	1.83	1.46
STL-AR	2.95	1.91	1.27	1.94	1.83

MASE of evolved series

Method	Yearly	Quarterly	Monthly	All
Naïve	1.51	2.32	1.64	1.82
Seasonal naïve	1.51	1.19	1.33	1.34
Theta	1.44	1.35	1.07	1.28
ETS	1.69	1.30	1.04	1.34
ARIMA	1.44	1.29	**0.93**	**1.22**
STL-AR	**1.41**	**1.07**	1.34	1.27

M3 is not a representative sample of any larger population of time series.

Comparison of forecasting methods



Conclusions

Conclusions

Identify unusual time series

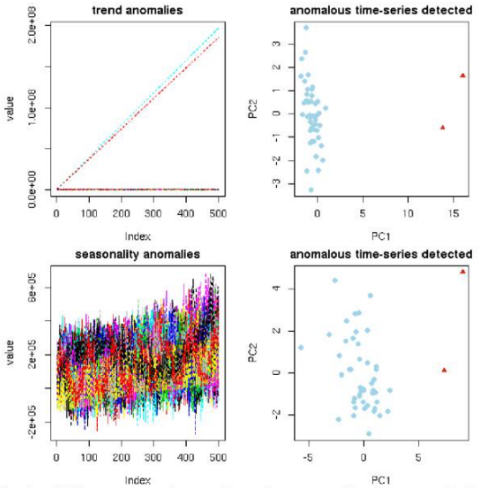
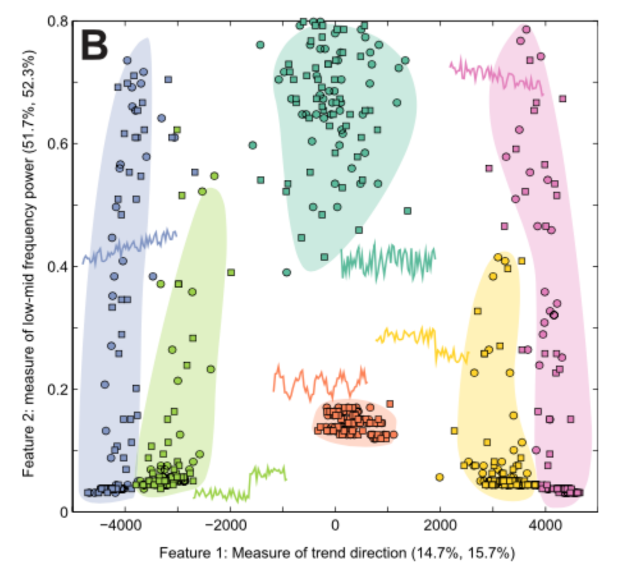


Fig. 1. Different types of anomalies and corresponding first two principal components which our method uses for unusual time series detection. These types of anomalous time series may be due to an abnormal server or a malicious user.

Find clusters



Find clusters

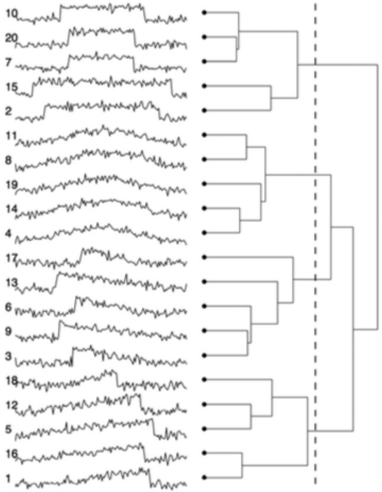


FIG. 4. Dendrogram from hierarchical clustering of the extracted shapes based on features; the vertical line shows where the binary tree is cut to get the four basic types of shapes.

Conclusions

- Generate new time series with specific features.
- M3 conclusions will not necessarily hold for other time series collections.
- Different forecasting methods perform better in some regions of the feature space than other methods.

What else can we do

- Develop meta-forecasting algorithms which choose a specific method based on the location of a time series in the instance space (almost done).
- Generate new time series with specific features with decent computation efficiency (almost done.)
- Develop **R** package **TSfeatures** which can extract thousands of features from a single time series (**to be done, now**).
- Select features automatically for a variety of tasks (**to be done**).
- Applications (**to be done**).

References

Yanfei Kang, Rob J. Hyndman, Kate Smith-Miles. Visualising forecasting algorithm performance using time series instance spaces. *International Journal of Forecasting* 33 (2), 345-358 (2017).

Ben D. Fulcher, Nick S. Jones. Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, 26(12), 3026-3037 (2014). Code available at https://github.com/benfulcher/hctsa.

http://yanfei.site

yanfeikang@buaa.edu.cn