# Demand forecasting by temporal aggregation: using optimal or multiple aggregation levels?

Nikolaos Kourentzes[a]

Bahman R. Tabar[b]

Devon K. Barrow[b]

[a]Lancaster University; [b]Coventry University

ISIR 2016

Lancaster University Management School | Lancaster Centre for Forecasting | Coventry Business School | Forecasting Society

# Agenda

1. Demand forecasting by temporal aggregation
2. Optimal aggregation level
3. Multiple aggregation levels
4. Evaluation

# How do we build models now?

- This is by no means a resolved question, but there are some reliable approaches. Key questions: model form & estimation.

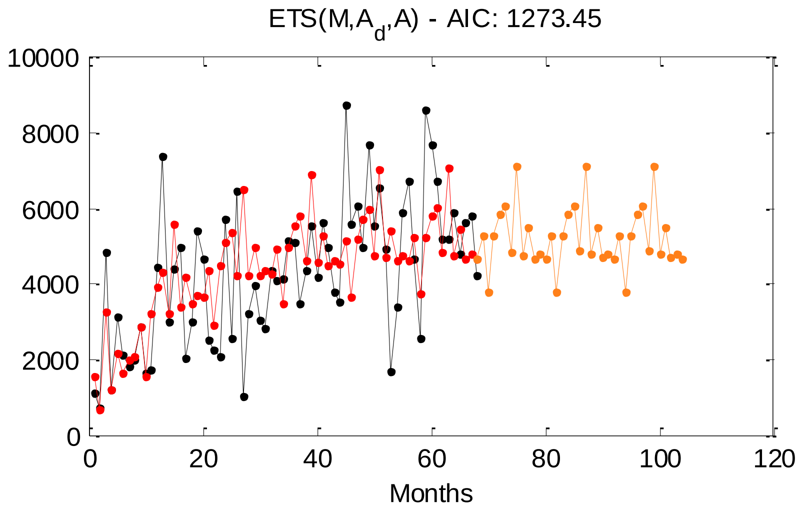
- Take the example of exponential smoothing family:
  - Considered one of the most reliable and robust methods for automatic univariate forecasting [Gardner, 2006].
  - It is a family of methods: **ETS (error type, trend type, seasonality type)** [Hyndman et al., 2002, 2008]
    - Error: **A**dditive or **M**ultiplicative
    - Trend: **N**one or **A**dditive or **M**ultiplicative, Linear or Damped/Exponential
    - Seasonality: **N**one or **A**dditive or **M**ultiplicative
  - Adequate for a most types of time series.
  - Within the state space framework we can select and fit model parameters automatically and reliably.

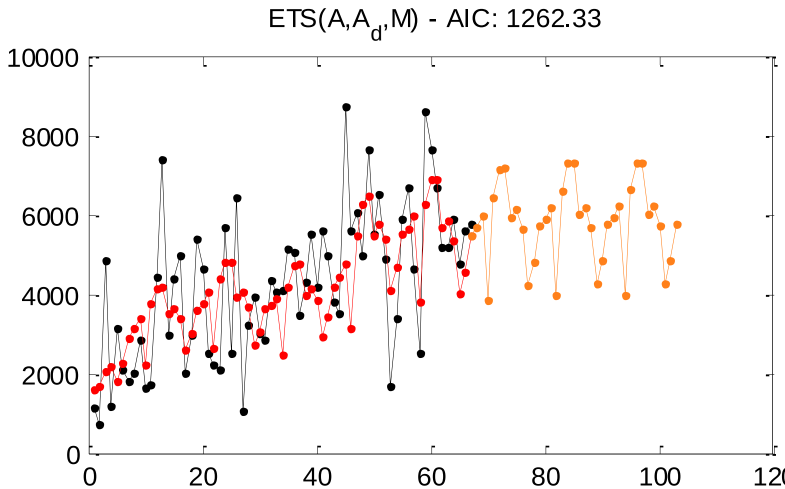
LCF

# Any issues with current forecasting practice?

Issues with modelling:

- Model selection → How good is the best fit model? How reliable?
- Sampling uncertainty → Identified model/parameters stable as new data appear?
- Model uncertainty → Appropriate model structure and parameters?
- Transparency/Trust → Practitioners do not trust systems that change substantially




A single additional observation changed the selected model and forecast!

# Temporal aggregation and forecasting

- Temporal aggregation has been explored as a way to help us deal with these issues.
- It is not new, but the question has been at **which single level to model the time series**. Econometrics have investigate the question for decades → inconclusive
- Supply chain applications: ADIDA [Nikolopoulos et al., 2011] → beneficial to slow and fast moving items forecast accuracy (like everything… not always!):
  - **Step 1**: Temporally aggregate time series to the appropriate level
  - **Step 2**: Forecast
  - **Step 3**: Disaggregate forecast and use
  - Selection of aggregation level → No theoretical grounding for general case, but good understanding for AR(1)/MA(1)/ARMA(1,1) cases [Rostami-Tabar et al., 2013, 2014].

LCF

# Temporal aggregation and forecasting

Recently there has been a resurgence in using temporal aggregation for forecasting.

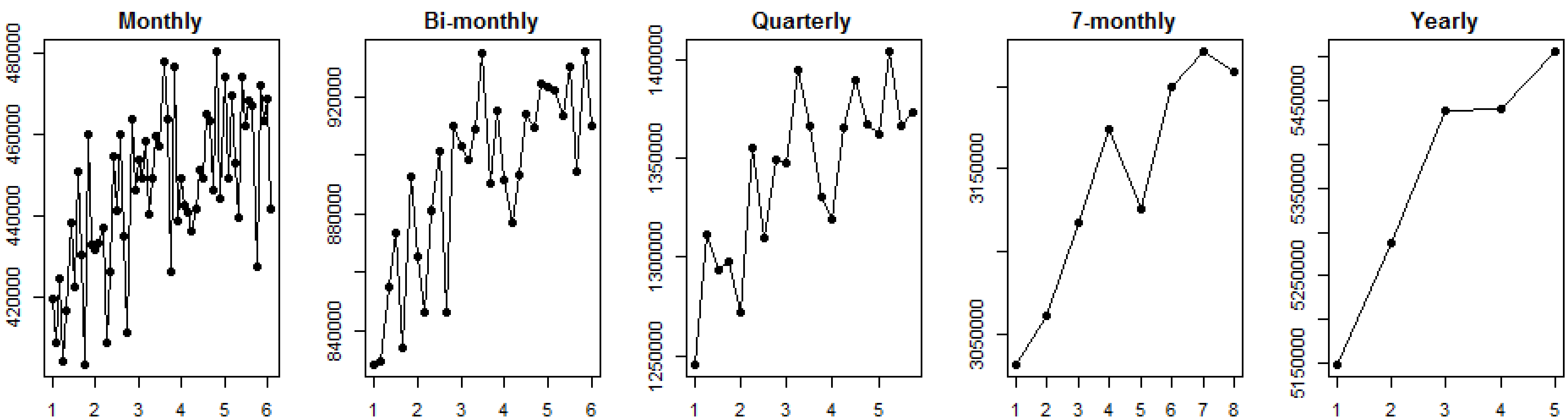
- Non-overlapping temporal aggregation is a moving average filter.
- Filters high frequency components: noise, seasonality, etc.
- Reduces intermittency [Nikolopoulos et al, 2011; Petropoulos & Kourentzes, 2014].
- But can increase complexity [Wei, 1978; Rossana & Seater, 1995; Silvestrini & Veredas, 2008]:
  - Loss of estimation efficiency;
  - Complicates dynamics of underlying (ARIMA) process;
  - Identifiable process converge to IMA - often IMA(1,1);
  - What is the best temporal aggregation to work on?

Two school of thoughts. **How do they compare?**

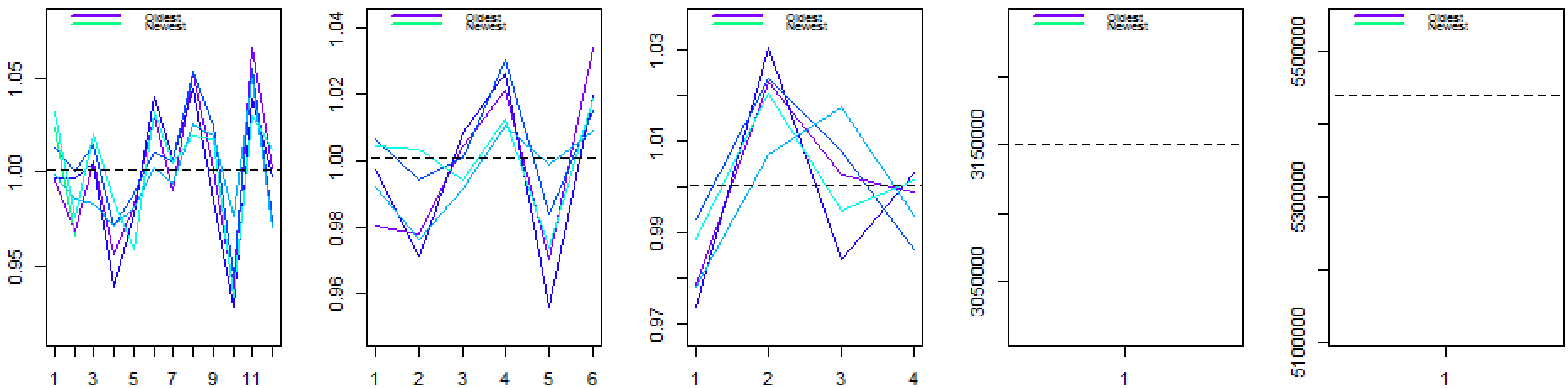
i. **Traditional**: Identify **a single optimal temporal aggregation level** to model [Rossana & Seater, 1995; Nikolopoulos et al., 2011; Rostami-Tabar et al., 2013, 2014].

ii. **Use multiple temporal aggregation levels** [Kourentzes et al., 2014; Kourentzes & Petropoulos, 2015].

# How temporal aggregation changes the series




Seasonal diagrams



# Identifying the optimal aggregation level

Rostami-Tabar et al. 2014 evaluate analytically the impact of temporal aggregation for **ARMA(1,1)**, **AR(1)** and **MA(1)** and derive formulas to find the optimal aggregation level in terms of MSE when forecasting with **Single Exponential Smoothing**.

$$\mathrm{MSE}_{ARMA} = \frac{2\sigma^2\left(k(1-2\phi\theta+\theta^2)+(\phi-\theta)(1-\phi\theta)\left(\sum_{i=1}^{k-1}2(k-i)\phi^{i-1}\right)\right)}{(2-\alpha)(1-\phi^2)} + \frac{2\sigma^2\alpha\left(\sum_{i=1}^{k}\left(i\phi^{(i-1)}\right)+\sum_{i=2}^{k}(i-1)\phi^{(2k-i)}\right)(\phi-\theta)(1-\phi\theta)}{(2-\alpha)(1-\phi^k+\alpha\phi^k)(1-\phi^2)}$$

$$\mathrm{MSE}_{AR} = 2\sigma^2\left(\frac{k+\sum_{i=1}^{k-1}2(k-i)\phi^i}{(1-\phi^2)(2-\alpha)}\right) - \frac{2\sigma^2\alpha\left(\sum_{i=1}^{k}\left(i\phi^{(i-1)}\right)+\sum_{i=2}^{k}(i-1)\phi^{(2k-i)}\right)}{(2-\alpha)(1-\phi^k+\alpha\phi^k)(1-\phi^2)}$$

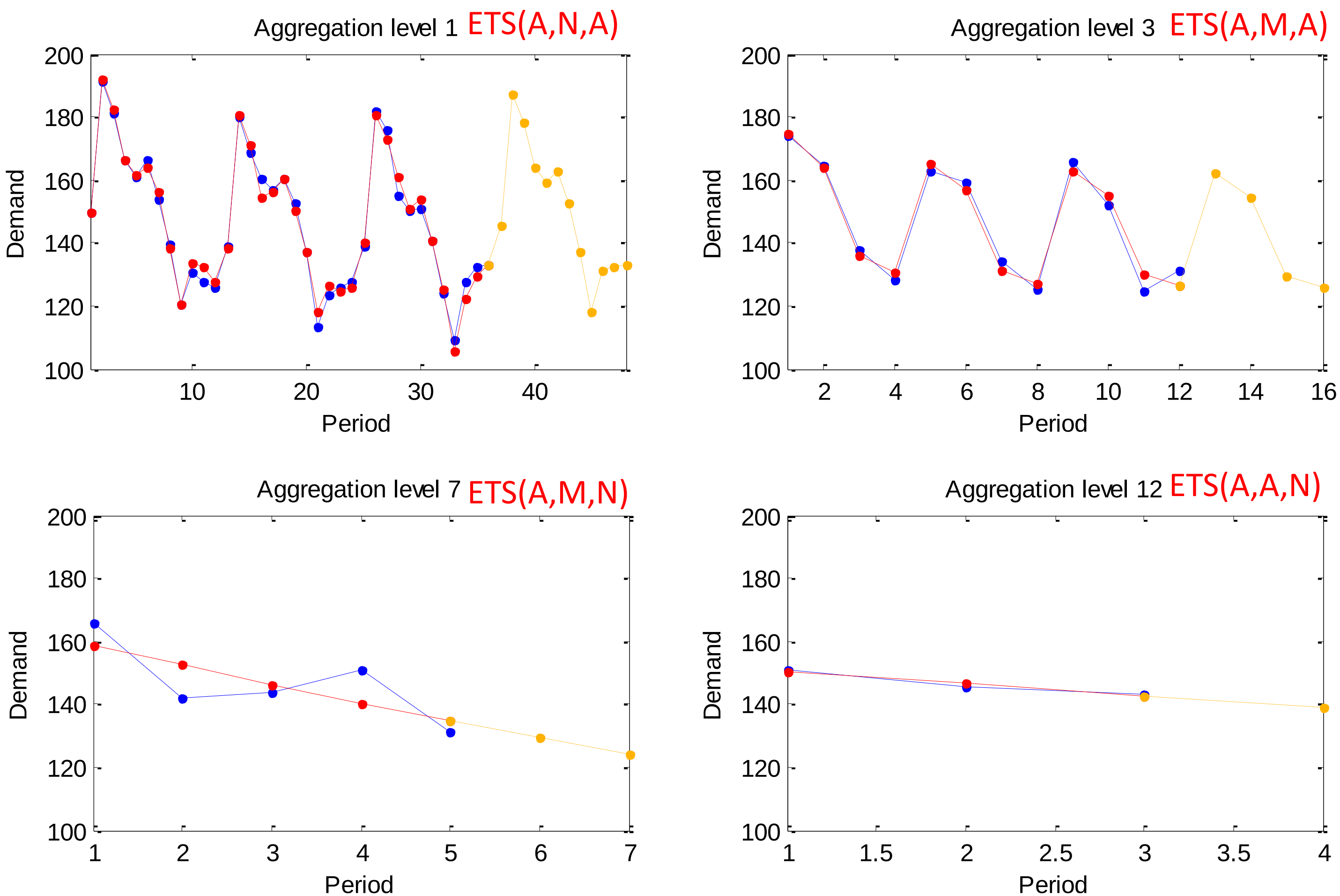
$$\mathrm{MSE}_{MA} = \frac{\sigma^2\left(2k(1+\theta^2)-2(k-1)\theta+2\alpha\theta\right)}{2-\alpha}$$

We need to know **parameters of original process (ϕ, θ)** and optimal smoothing **parameter α for SES** at **aggregate level k**.

Calculate MSE for various k and pick the best to find the optimum temporal aggregation level.

# Using multiple aggregation levels

What if we do not select an aggregation level? → use multiple [Kourentzes et al., 2014]

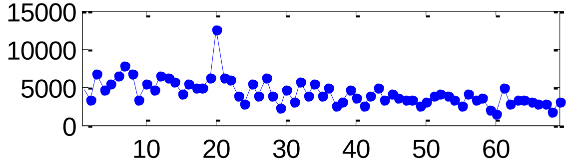



**Issues**:

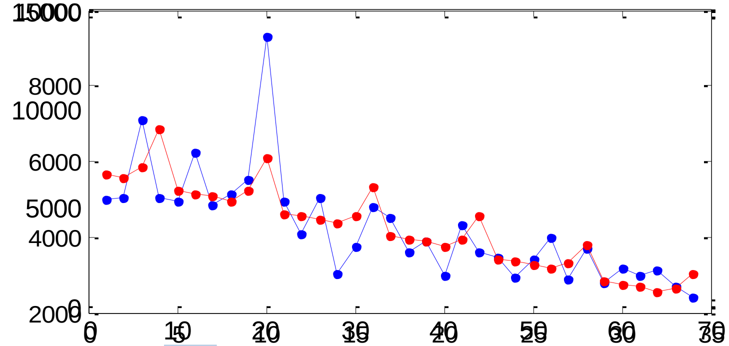
- Different model
- Different length
- Combination

## Aggregate

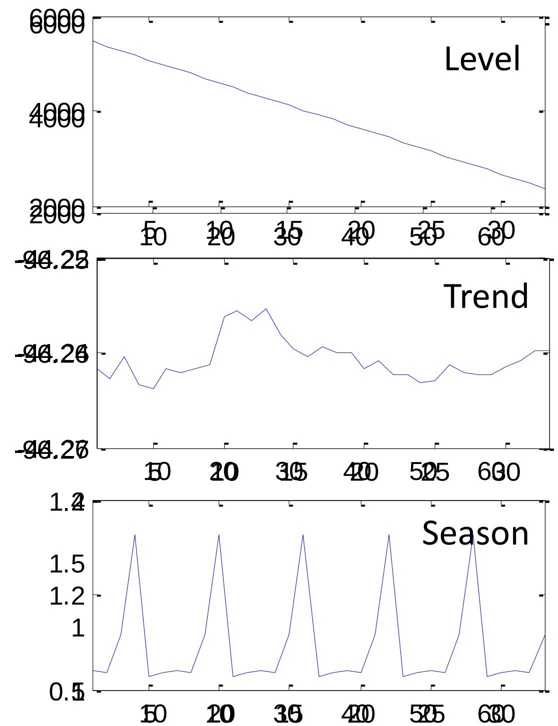
$y_t^{[1]}$



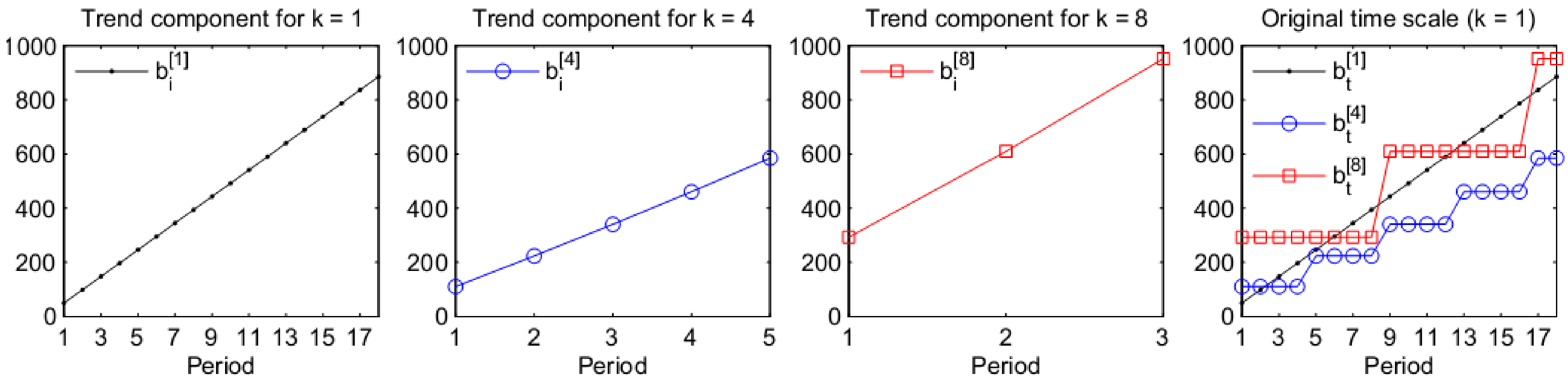
## Fit state space ETS



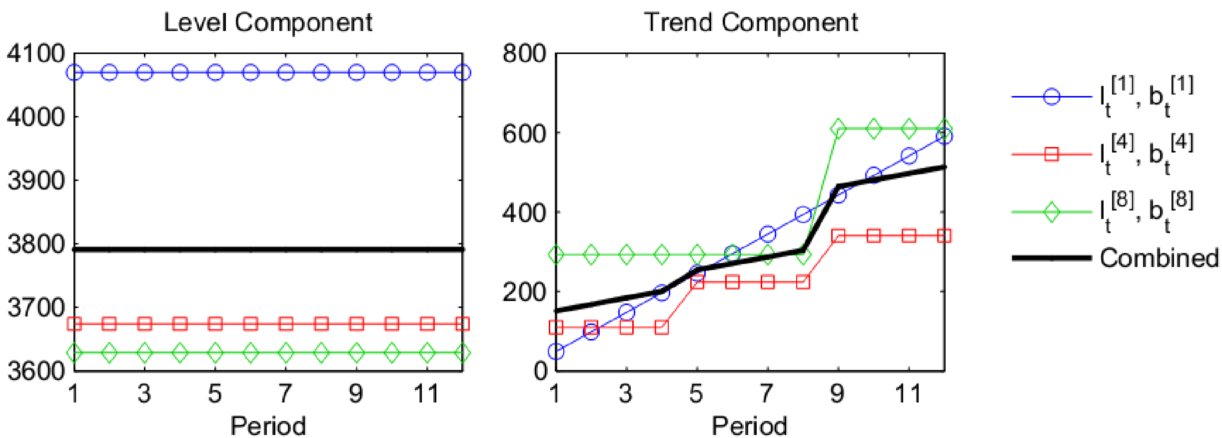
## Save states



## Transform states to additive and to original sampling frequency




## Combine states (components)

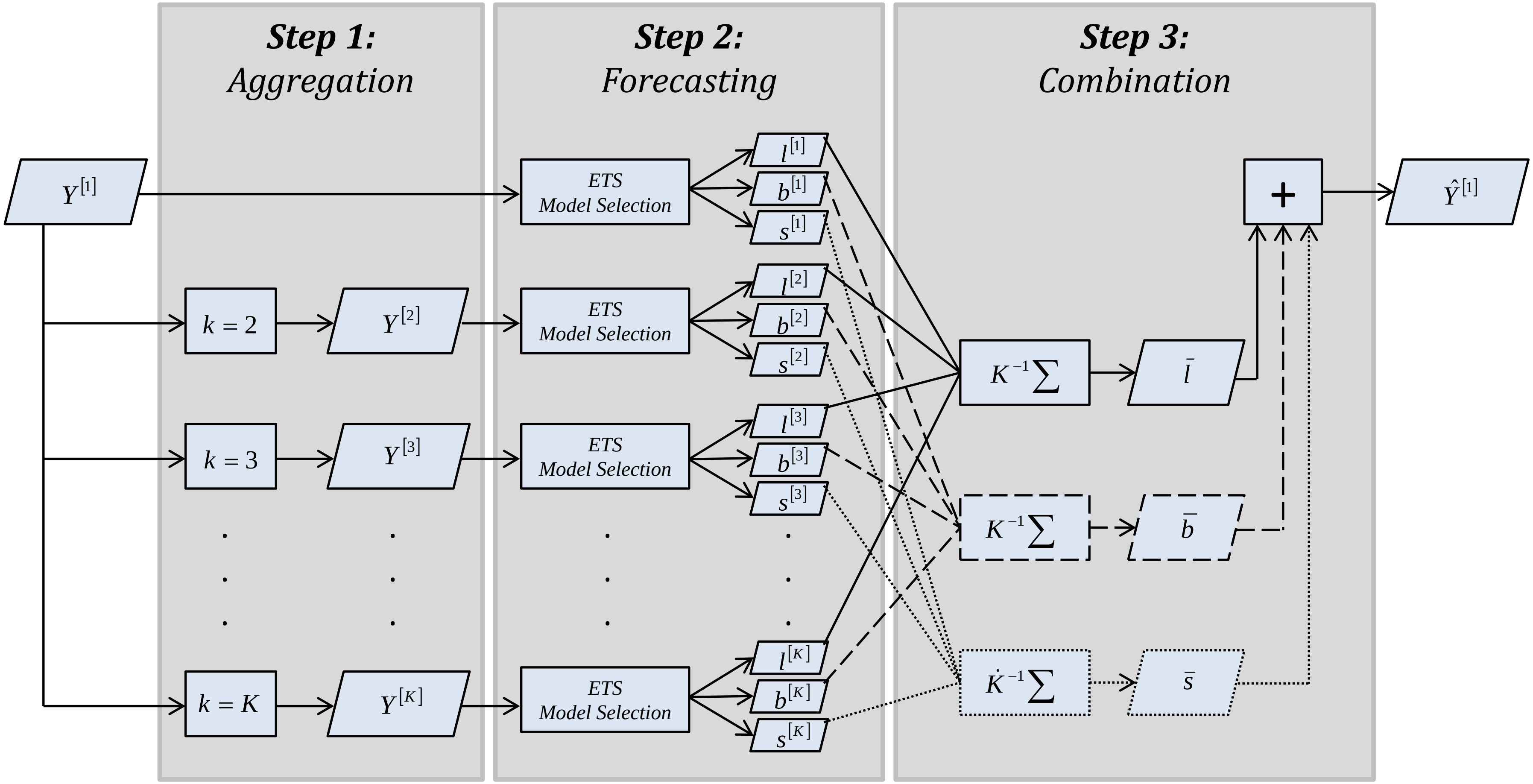



## Produce forecasts

$$\hat{y}^{[1]}_{t+h^{[1]}} = \bar{l}_{t+h^{[1]}} + \bar{b}_{t+h^{[1]}} + \bar{s}_{t-S+h^{[1]}}$$

LCF

# Multiple Aggregation Prediction Algorithm (MAPA)




Strengthens and attenuates components

Estimation of parameters at multiple levels

Robustness on model selection and parameterisation

LCF

# Empirical evaluation

- **Forecast the next 13 periods**
  - **Simulated**: known processes can be used to assess the optimal selection
    - ARIMA(p,d,q), with p = (0,1,2), d = (0,1) and q = (0,1,2)
    - 500 series each process, 60 fit set & 40 test set.
  - **Real**: 229 series of 173 weekly observations – non-seasonal
    - ADF test suggests that 90% is ARIMA(p,0,q) and 10% ARIMA(p,1,q).
    - 130 fit set & 43 test set
- Accuracy metric: ARMAE; < 1 better than benchmark!

$$\mathrm{MAE} = m^{-1} \sum_{t=1}^{m} |y_t - \hat{y}_t| \qquad \mathrm{ARMAE} = \sqrt[n]{\prod \left( \frac{\mathrm{MAE}_i}{\mathrm{MAE}_b} \right)}$$

# Methods

- **Original sampling frequency (no aggregation)**
  - **Benchmark**: Single exponential smoothing – **Orig-SES**
  - ETS model family – **Orig-ETS**
- **Single temporal aggregation**
  - Heuristic based level (13 periods) – **Heur-SES**
  - Optimal level – **Opt-SES**
- **Multiple temporal aggregation**
  - Restricted to SES only – **MAPA-SES**
  - Unrestricted – **MAPA**

LCF

# Accuracy - ARMAE

| Demand | No aggregation | | Single level | | | Multiple levels | |
|---|---|---|---|---|---|---|---|
| | Orig-SES | Orig-ETS | Heur-SES | | Opt-SES | MAPA-SES | MAPA |
| ARIMA(1,0,0) | 1.000 | 0.979 | 0.974 | < | 0.975 | 0.972 | **0.961** |
| ARIMA(0,0,1) | 1.000 | 1.002 | **0.960** | < | 0.965 | 0.972 | 0.973 |
| ARIMA(2,0,0) | 1.000 | 0.971 | 0.986 | > | 0.983 | 0.973 | **0.949** |
| ARIMA(0,0,2) | 1.000 | 1.002 | **0.969** | = | **0.969** | 0.978 | 0.979 |
| ARIMA(1,0,1) | 1.000 | 1.001 | 0.966 | < | 0.971 | 0.964 | **0.963** |
| ARIMA(2,0,2) | 1.000 | 0.983 | 0.990 | > | 0.982 | 0.974 | **0.953** |
| ARIMA(1,1,0) | **1.000** | **1.000** | 1.439 | > | 1.223 | 1.062 | 1.004 |
| ARIMA(0,1,1) | **1.000** | 1.051 | 1.290 | > | 1.173 | 1.030 | 1.037 |
| ARIMA(2,1,0) | 1.000 | **0.891** | 1.444 | > | 1.207 | 1.062 | 0.916 |
| ARIMA(0,1,2) | **1.000** | 1.048 | 1.278 | > | 1.091 | 1.011 | 1.012 |
| ARIMA(1,1,1) | 1.000 | **0.975** | 1.349 | > | 1.191 | 1.056 | 0.990 |
| ARIMA(2,1,2) | 1.000 | 0.927 | 1.327 | > | 1.139 | 1.044 | **0.922** |
| ARIMA(∗,0,∗) | 1.000 | 0.989 | 0.974 | = | 0.974 | 0.972 | **0.963** |
| ARIMA(∗,1,∗) | 1.000 | 0.980 | 1.353 | > | 1.170 | 1.044 | **0.979** |
| ARIMA(∗,∗,∗) | 1.000 | 0.985 | 1.148 | > | 1.068 | 1.007 | **0.971** |
| Real Data | 1.000 | 1.011 | 0.999 | = | 0.999 | **0.992** | 0.994 |

# Conclusions

- Temporal aggregation **improves accuracy → use it**!
- Identifying **optimal aggregation level** does not always work as expected, but **overall equal if not better than heuristic** for selecting level.
- Why? **Optimal selection** of level is **not robust enough to model uncertainty** at the original and aggregate level → we simply do not know the true process and optimal selection assumes knowledge of it.
- **MAPA** by construction is **suboptimal** for any process, **but it is very robust** and reliable → consistently resulted in better accuracy (matches the literature).
- Future research should focus on:
  - **How can we make optimal more robust?**
  - **How can we make MAPA "more optimal"?**

LCF

# Temporal aggregation R code!

To temporally aggregate a series use the function `tsaggr` from the **MAPA** package:
http://cran.r-project.org/web/packages/MAPA/index.html

Code for finding the **optimal temporal aggregation level** is available for R, in the **TStools** package, which is available at GitHub (not in CRAN yet):
https://github.com/trnnick/TStools - function: `get.opt.k`

The **Multiple Aggregation Prediction Algorithm** is available for R, in the **MAPA** package:
http://cran.r-project.org/web/packages/MAPA/index.html
Its intermittent demand counterpart is available in the **tsintermittent** package:
http://cran.r-project.org/web/packages/tsintermittent/index.html
Examples and interactive demos for both are available at my blog:
http://nikolaos.kourentzes.com

LCF

# Thank you for your attention!
# Questions?

Published, working papers and code available at my blog!

Nikolaos Kourentzes

email: n.kourentzes@lancaster.ac.uk

blog: http://nikolaos.kourentzes.com

Lancaster University Management School | Lancaster Centre for Forecasting