An Algorithm for Forecasting Time Series From the M4 Competition Dataset

David Veitch

University of Toronto

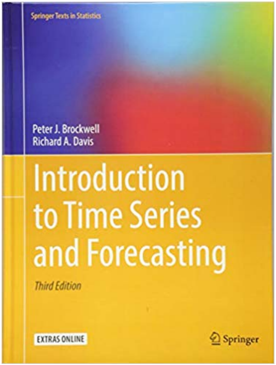
http://daveveitch.github.io

August 2019

Agenda

1. Introduction
   - Project Origin
   - The M4 Dataset
   - Challenges of the Competition
2. Time Series Models
3. Measuring Performance
4. Modelling Process
5. Computation
6. Performance
7. Appendix

Project Origin


Springer Texts in Statistics
Peter J. Brockwell
Richard A. Davis
Introduction to Time Series and Forecasting
Third Edition
EXTRAS ONLINE
Springer

- Summer reading course on time series analysis with Professor Zhou Zhou
- Cumulative project (this presentation and report) an opportunity to apply what I learnt
- Goal to create an accurate and interpretable forecasting method for the M4 Competition dataset

The M4 Dataset

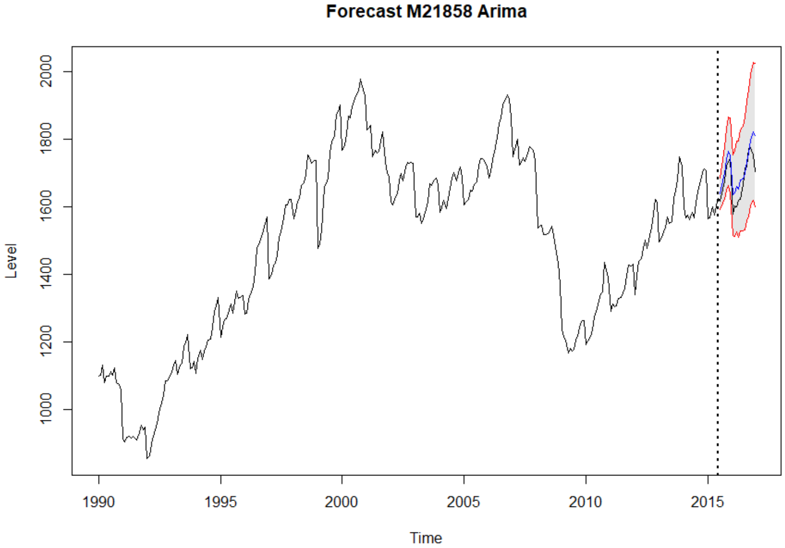
M4COMPETITION

Fourth competition

Fourth competition, started in Jan 1 2018, ended in May 31 2018.

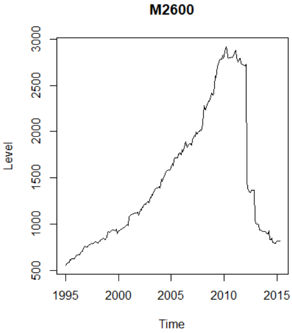
- `https://www.mcompetitions.unic.ac.cy/`
- Fourth iteration of the Makridakis time series forecasting competition
- 100,000 time series from economics, demographics, etc. at various frequencies
- Competitors compute point forecasts and prediction intervals for each series
- M3 had only 3,003 time series, no prediction intervals, no machine learning benchmark methods

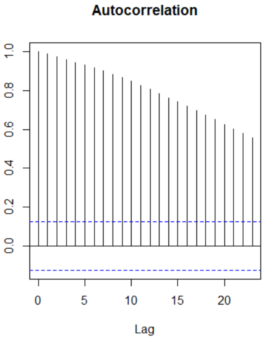
Example Time Series & Forecast



Challenges of the Competition

- **Blind Forecasting**
  - Variable Transformations
  - Model Selection
- **Dataset Size**
  - I only consider monthly time series; leaves 48,000 time series
  - Erratic behaviour
  - High computational cost





Time Series Models

Four models from the ARIMA family were used:

- ARIMA
- Holt Winters'
- Random Walk
- Random Walk with Drift

Model Selection Criteria

$$AICC = AIC + \frac{2(m+1)(m+2)}{n-m-2}$$
$$AIC = -2\log \hat{L} + 2m$$

- $\hat{L}$ the likelihood of the data under the model
- n the number of observations
- m the number of parameters in the model, $m = p + q + P + Q + k$ for a seasonal ARIMA model
- $k = 1$ if the model contains drift, $k = 0$ otherwise

Time Series Models - 1. ARIMA

Advantage: Can represent nonstationary series well
Disadvantage: Can potentially overfit the data.

$$\begin{aligned}\phi(B)\Phi(B^s)Y_t &= \theta(B)\Theta(B^s)Z_t \\ Y_t &= (1-B)^d(1-B^s)^D X_t \\ Z_t &\sim \mathrm{WN}(0,\sigma^2)\end{aligned}$$

Example: The ARIMA(1,1,0)x$(0,1,0)_{12}$ model would be:

$$\begin{aligned}Y_t &= (X_t - X_{t-12}) - (X_{t-1} - X_{t-13}) \\ Y_t &= \phi Y_{t-1} + Z_t \\ Z_t &\sim \mathrm{WN}(0,\sigma^2)\end{aligned}$$

Time Series Models - 2. Holt Winters'

Advantage: Nice interpretation as trend and seasonal components
Disadvantage: Just a limited version of a more general ARIMA model.

$$\begin{aligned}
\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t+h-m(k+1)} \\
\ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \\
s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}
\end{aligned}$$

- ℓ_t trend level
- b_t trend slope
- s_t seasonal components

Time Series Models - 3. Random Walk

Advantage: Many processes do exhibit random walk behaviour
Disadvantage: Very naïve model

$$
\begin{aligned}
X_t - X_{t-1} &= Z_t \\
Z_t &\sim \mathrm{IID}(0, \sigma^2)
\end{aligned}
$$

Time Series Models - 4. Random Walk With Drift

Advantage: Many processes do exhibit random walk with drift behaviour
Disadvantage: Very naïve model

$$
\begin{aligned}
X_t - X_{t-1} &= \mu + Z_t \\
Z_t &\sim \mathrm{IID}(0, \sigma^2)
\end{aligned}
$$

- μ the average change between periods

Measuring Performance

Point Forecasts: Two measures are calculated. The performance of my model is then compared against the performance of a naïve method (provided by the M4 competition). An average is then taken for the ultimate performance measure Overall Weighted Average (OWA).

$$\text{sMAPE} = \frac{1}{h}\sum_{i=1}^{h}\frac{2|Y_i - \hat{Y}_i|}{|Y_i| + |\hat{Y}_i|}$$

$$\text{MASE} = \frac{1}{h}\frac{\sum_{i=1}^{h}|Y_i - \hat{Y}_i|}{\frac{1}{n-m}\sum_{i=m+1}^{n}|Y_i - Y_{i-m}|}$$

Prediction Intervals: The M4 competition uses a specific performance measure called MSIS. For my purposes I just look at what percentage of actual values lie within my 95% prediction intervals.

Modelling Process

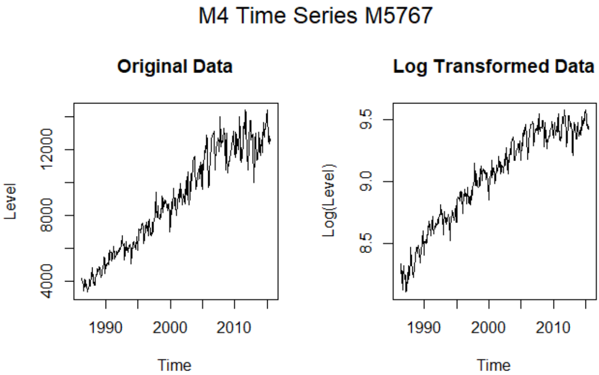
1. Heteroscedasticity Detection
2. Changepoint Detection
3. Model Fitting
4. Individual Forecasts
5. Forecast Averaging

Modelling Process - 1. Heteroscedasticity Detection

Test inspired by the Breusch-Pagan Test.

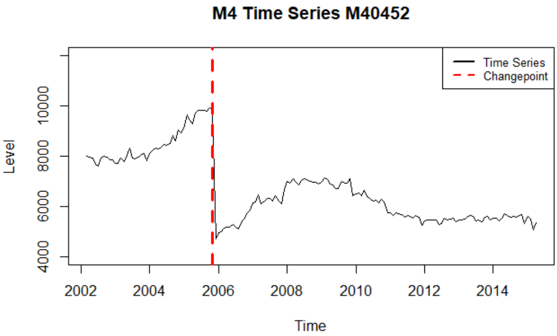
$$\hat{u}^2 = \beta_0 + \beta_1 FittedValues + \epsilon$$

Progressively stronger transformations until variance of residuals not increasing with level of time series.



Modelling Process - 2. Changepoint Detection

The At Most One Changepoint method (AMOC) looks for areas where the time series' behaviour changes in a significant way.



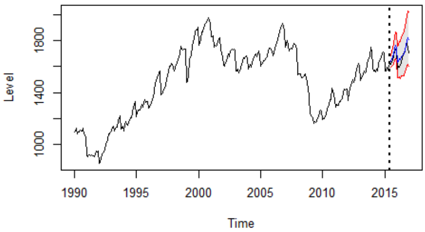
$$\tau_1 = \max_{\tau} \log p(y_{1:\tau}|\hat{\theta}_1) + \log p(y_{(\tau+1):n}|\hat{\theta}_2)$$

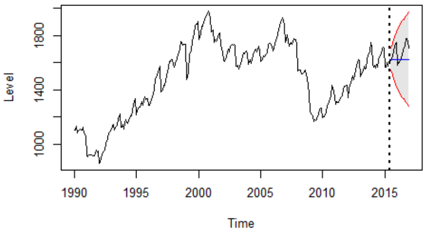
$$\lambda = 2\Big[\big(\log p(y_{1:\tau_1}|\hat{\theta}_1) + \log p(y_{(\tau_1+1):n}|\hat{\theta}_2)\big) - \log p(y_{1:n}|\hat{\theta})\Big]$$

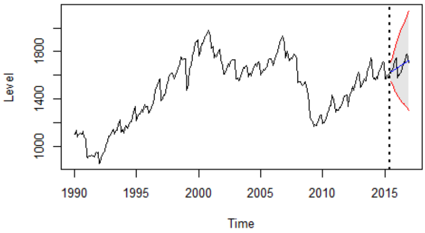
Modelling Process - 3. Model Fitting

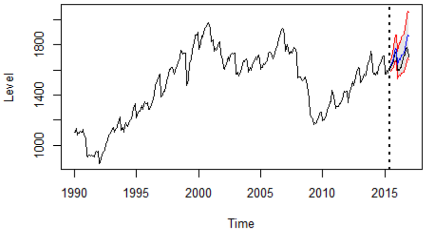
Select optimal parameters for each model via AICC maximization.

Modelling Process - 4. Individual Forecasts


Forecast M21858 Arima


Forecast M21858 RW


Forecast M21858 RWDrift


Forecast M21858 HW

Modelling Process - 5. Forecast Averaging

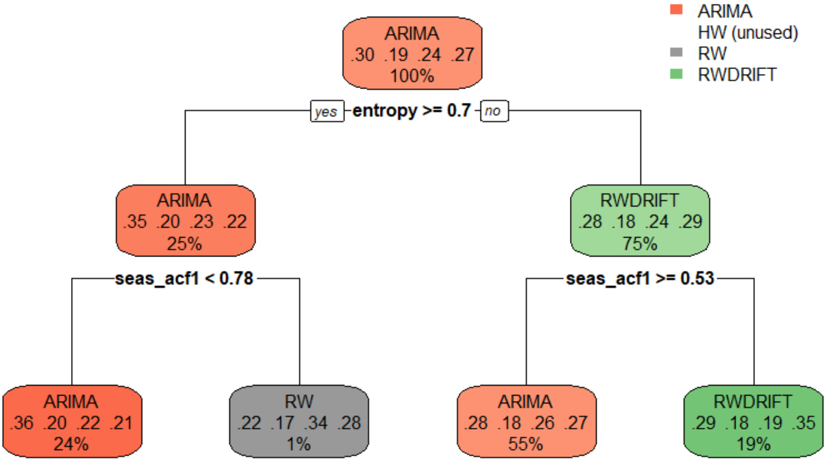
1. Utilize decision tree to determine optimal way to weight forecasts
2. Decide which model to use based on time series' features, and training set performance
3. Calculate probability each model is the best model for a given time series
4. Weight forecasts using these probabilities.

Features used include autocorrelation, seasonal autocorrelation and spectral entropy (a measure of how chaotic a time series is).

$$\text{Spectral Entropy} = -\int_{-\pi}^{\pi} \hat{f}(\lambda) \log \hat{f}(\lambda) d\lambda$$

Where $\hat{f}(\lambda)$ is the estimated spectral density of the data.

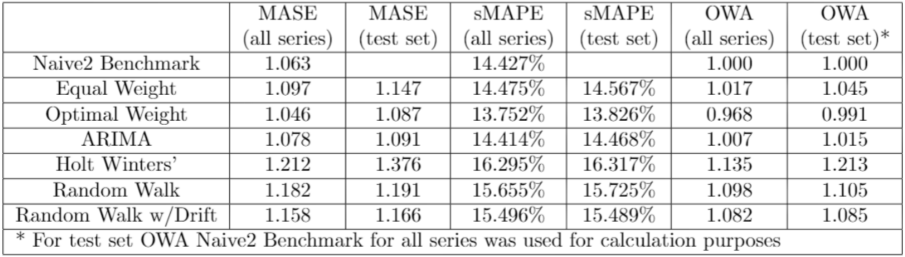
Modelling Process - 5. Forecast Averaging



Computation

1. Need to use parallel computing to compute forecasts for the 48,000 time series
2. R's `foreach` and `doParallel` packages in combination with 20 cores on the UofT Statistics server

Performance - Absolute

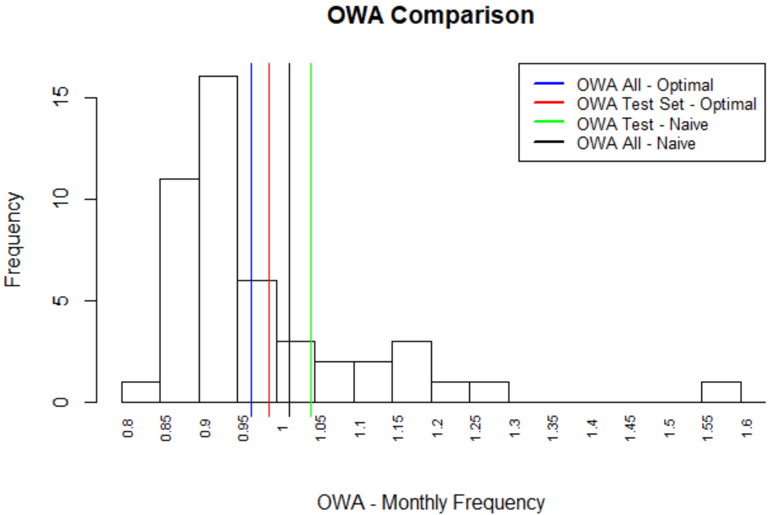


	MASE (all series)	MASE (test set)	sMAPE (all series)	sMAPE (test set)	OWA (all series)	OWA (test set)*
Naive2 Benchmark	1.063		14.427%		1.000	1.000
Equal Weight	1.097	1.147	14.475%	14.567%	1.017	1.045
Optimal Weight	1.046	1.087	13.752%	13.826%	0.968	0.991
ARIMA	1.078	1.091	14.414%	14.468%	1.007	1.015
Holt Winters'	1.212	1.376	16.295%	16.317%	1.135	1.213
Random Walk	1.182	1.191	15.655%	15.725%	1.098	1.105
Random Walk w/Drift	1.158	1.166	15.496%	15.489%	1.082	1.085

* For test set OWA Naive2 Benchmark for all series was used for calculation purposes

- Large benefit from using optimal weights than naive, or any single method
- Holt Winters' does a surprisingly poor job
- 95% prediction intervals contain out-of-sample values between 95.21%-95.45% of the time

Performance - Relative



Future Directions/Other Thoughts

Future Directions

- Increase number of forecasting methods (including more ML)
- What is the optimal feature set for describing a time series?

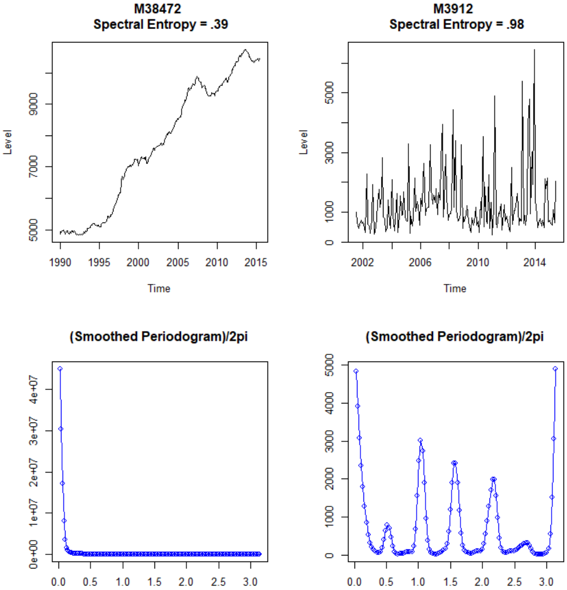
Other Thoughts

- Are ‘blind forecasting’ competitions like this even that valuable?
- Machine learning was instrumental in the M4 winners’ submission, as well as helping improve my own. What is the best way to leverage ML in the time series setting?

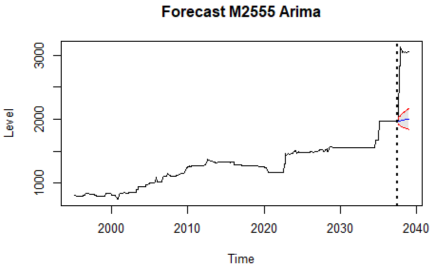
Thank you!

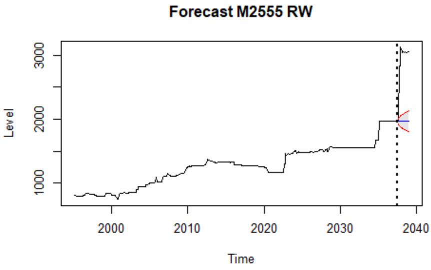
Appendix

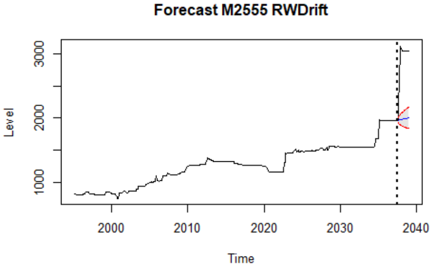
Spectral Entropy

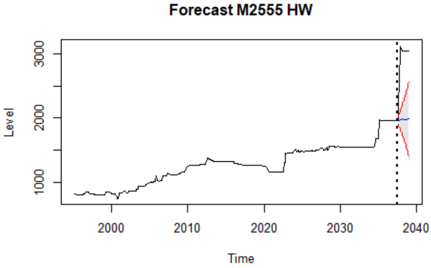


Bad Forecast Example


Forecast M2555 Arima


Forecast M2555 RW


Forecast M2555 RWDrift


Forecast M2555 HW