Feature-based time series generation and forecasting

Feng Li
Central University of Finance and Economics
http://feng.li/

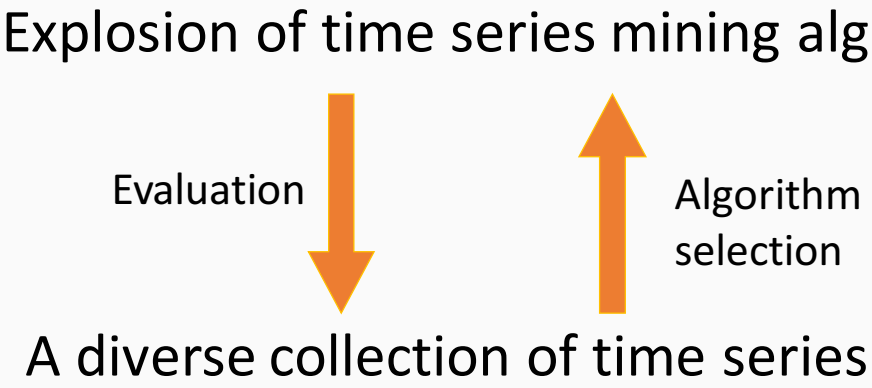
— *based on a series of working papers:*

* Efficient generation of time series with diverse and controllable characteristics
(with Yanfei Kang and Rob Hyndman)
* Forecasting using time series feature spaces
(with Yanfei Kang and Rob J. Hyndman)
* Time series forecasting based on automatic feature extraction
(with Yanfei Kang, Xixi Li)

Motivation

Motivation

- Train a time series model (*machine learning with dependent data*) is usually costly.
- New algorithms are developed every day.



- Why not cloud computing? Privacy is the main concern in practice.
- A well trained model with my dataset does not necessary work well for your dataset. Why?
- Is there an efficient way to **forecast which algorithm works the best** for a particular time series.

The No-Free-Lunch theorem

- There is **never** universally best method that fits in all situations (Wolpert, 1996).
- The **explosion of new algorithms** development makes the question even more worth focusing.
- No single forecasting method stands out the best for any type of time series (Adam, 1973).

Literature

Literature

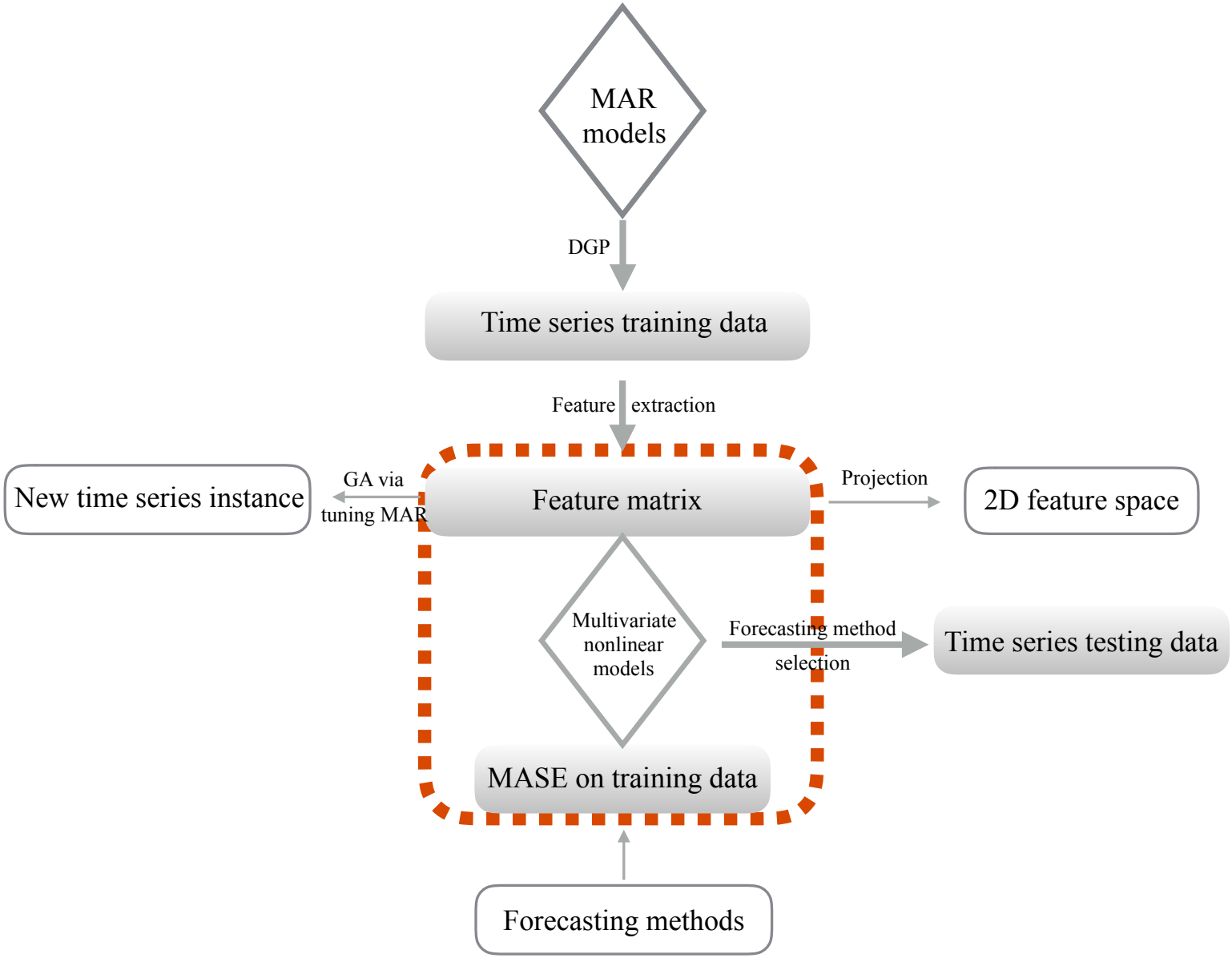
- Features of time series $\rightarrow$ benefits in producing more accurate forecasting accuracies (Adam, 1973).
- Features $\rightarrow$ forecasting method selection rules (Meade, 2000).
- "Horses for courses" $\rightarrow$ effects of time series features to the forecasting performances (Petropoulos et al., 2014).
- We could visualize the performances of different forecasting methods in a 2D space $\rightarrow$ to get better understanding of their relative performances (Kang et al., 2017).

Existing problems

- For a collection of time series, a selection process can hardly beat **single-method** forecasting strategies.
  - select a good, but not necessarily the best method (Meade, 2000).
  - impossible to identify one single optimal method based on features (Petropoulos et al., 2014).
  - space can not be separated nicely according to the strengths and weaknesses of forecasting methods (Kang et al., 2017).
- Possible reasons?
  - Inadequate time series features
  - limited training time series data (not only in number, but in diversity)

What we do?

The framework



Final aim: forecasting on testing data

- **Training data**: Randomly generated data via our DGP procedure.
- **Testing data**: M3 data (Makridakis and Hibon, 2000)
  - 3003 time series
  - From demography, finance, business and economics
  - Lengths between 14 and 126
  - Either non-seasonal, monthly or quarterly
  - Positive

Time series features

Basic idea

Transform a given time series $\{x_1, x_2, \cdots, x_n\}$ to a feature vector $F = (F_1, F_2, \cdots, F_p)'$ (Kang et al., 2017; Hyndman et al., 2015)

A feature F_k can be any kind of function computed from a time series:

1. A simple mean
2. The parameter of a fitted model
3. Some statistic intended to highlight an attribute of the data
4. ...

Which features should we use?

- There does not exist the best feature representation of a time series (Fulcher, 2018).
- Depends on both the **nature** of the time series being analysed, and the **purpose** of the analysis.
  - With unit roots, the mean is not a meaningful feature without some constraints on the initial values.
  - CPU usage every minute for a large number of servers: we observe a daily seasonality. The mean may provide useful comparative information despite the time series not being stationary.

Which features should we use?

- Time series are of different lengths, on different scales, and with different properties.
- We restrict our features to be ergodic, stationary and independent of scale.
- 17 sets of diverse features.
- New features are intended to measure attributes associated with multiple seasonality, non-stationarity and heterogeneity of the time series.

Which features should we use?

- Or automatic feature extraction
  - We introduce an automated approach to extract time series features based on images
  - Time series are first transformed into recurrence images, from which local features can be extracted using computer vision algorithms.
  - The extracted features are used for forecast model selection and model averaging.

Multiple seasonal time series

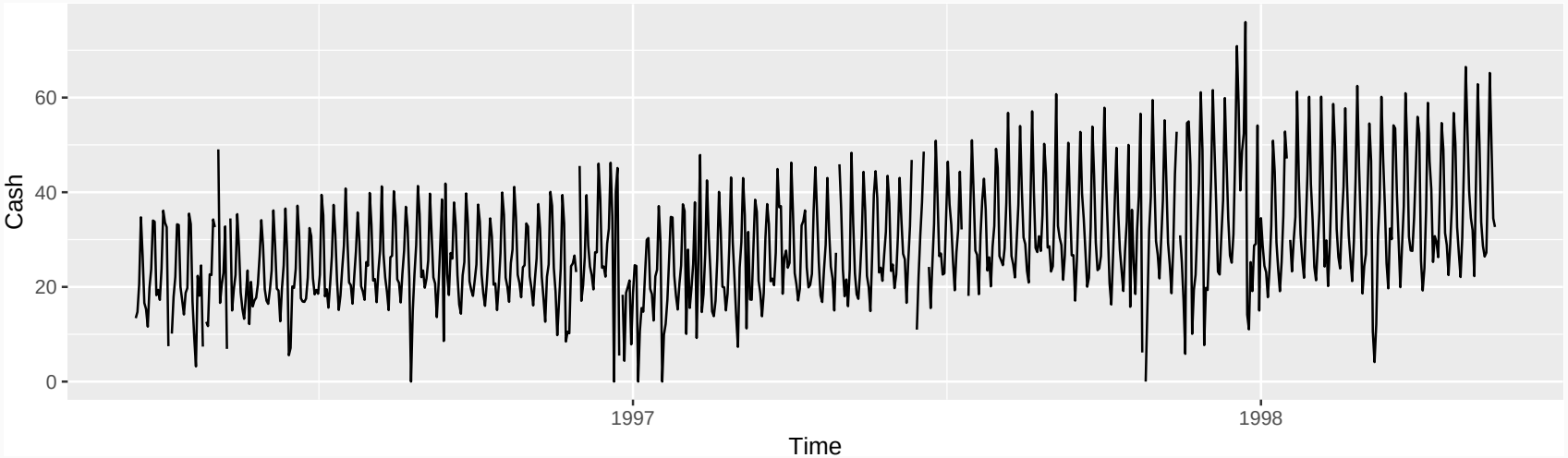


Figure 1: Daily cash money demand at some automatic teller machine.

STL decompostion extension

$$x_t = f_t + s_{1,t} + s_{2,t} + \cdots + s_{M,t} + e_t.$$

The strength of trend can be measured by:

$$F_{10} = 1 - \frac{\text{var}(e_t)}{\text{var}(f_t + e_t)}.$$

The strength of seasonality for the ith seasonal component:

$$F_{11,i} = 1 - \frac{\text{var}(e_t)}{\text{var}(s_{i,t} + e_t)}.$$

Features on heterogenity

1. Pre-whiten the time series x_t to remove the mean, trend, and Autoregressive (AR) information.
2. Fit an GARCH(1,1) model on the pre-whitened time series y_t to measure for the ARCH effects.
3. Test for the arch effects in the obtained residuals z_t using a second GARCH(1,1) model.

Features on heterogenity

The features of time series on heterogenity.

1. The sum of squares of the first 12 autocorrelations of $\{y_t^2\}$.
2. The sum of squares of the first 12 autocorrelations of $\{z_t^2\}$.
3. The R^2 value of an AR model applied to $\{y_t^2\}$.
4. The R^2 value of an AR model applied to $\{z_t^2\}$.

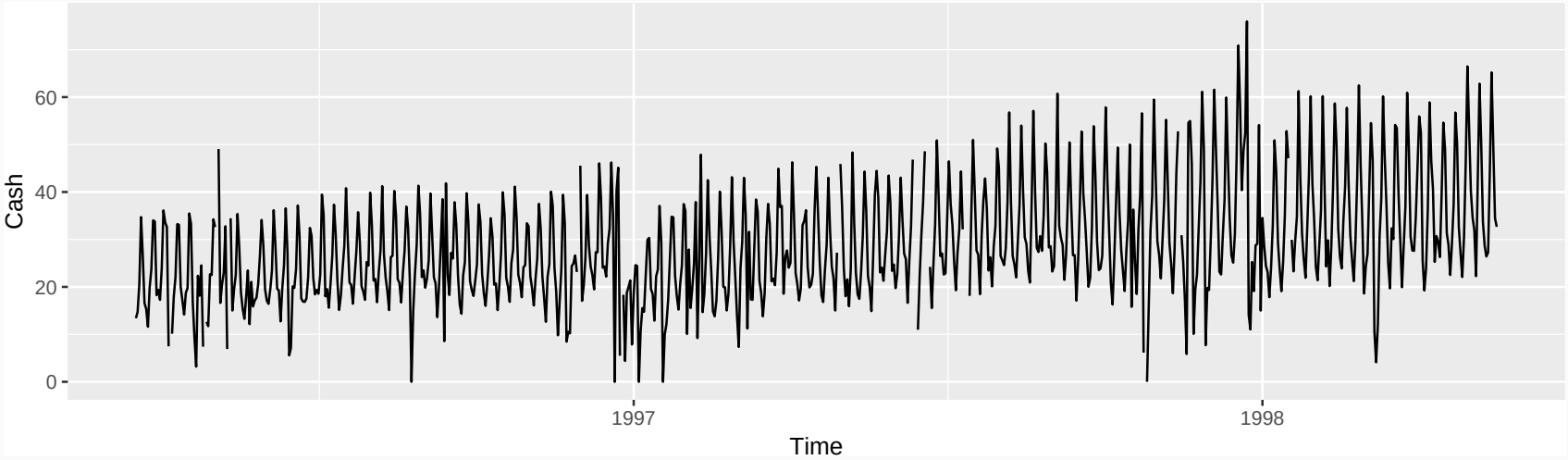
Time series features we use

Feature	Description	Feature	Description
F_1	Number of seasonal periods	F_{10}	Strength of trend
F_2	Vector of seasonal periods	F_{11}	Strength of seasonality
F_3	Number of differences for stationarity	F_{12}	Spikiness
F_4	Number of seasonal differences for stationarity	F_{13}	Autocorrelation coefficients of remainder
F_5	Autocorrelation coefficients	F_{14}	ARCH ACF statistic
F_6	Partial autocorrelation coefficients	F_{15}	GARCH ACF statistic
F_7	Spectral entropy	F_{16}	ARCH R^2 statistic
F_8	Nonlinearity coefficient	F_{17}	GARCH R^2 statistic
F_9	Long-memory coefficient		

- R package: `tsfeatures` available on GitHub

  https://github.com/robjhyndman/tsfeatures

A multi-seasonal time series with extracted features



entropy	x.acf1	x.acf10	diff1.acf1	diff1.acf10	diff2.acf1
0.7647817	0.517701	1.305203	-0.03341881	0.6712913	-0.4276741
diff2.acf10	seas.acf1	x.pacf5	diff1x.pacf5	diff2x.pacf5	seas.pacf
0.4801166	0.1024061	0.4219687	0.5021309	0.5093377	0.01495473
nperiods	seasonal.period1	seasonal.period2	trend	spike	
2	7	365	0.6505438	1.094574e-07	
linearity	curvature	e.acf1	e.acf10	seasonal.strength1	seasonal.strength2
12.2213	1.430082	0.2235961	0.07897381	0.8365591	0.6138975
peak1	peak2	trough1	trough2	ndiffs	nsdiffs
3	357	7	359	1	1
hurst	nonlinearity	arch.acf	garch.acf	arch.r2	garch.r2
0.8418821	0.122436	0.3669286	0.02484353	0.2535115	0.02558531

Training data simulation

Gaussian Mixure Autoregressive (MAR) models

- Consist of multiple stationary or non-stationary autoregressive components.
- A K-component MAR model is defined as (Wong and Li, 2000) :

$$F(x_t|\mathcal{F}_{t-1}) = \sum_{k=1}^{K} \alpha_k \Phi\left(\frac{x_t - \phi_{k0} - \phi_{k1}x_{t-1} - \cdots - \phi_{kp_k}x_{t-p_k}}{\sigma_k}\right),$$

where $F(x_t|\mathcal{F}_{t-1})$ is the conditional cumulative distribution of x_t give the past information $\mathcal{F}_{t-1}$. $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. $\sum_{k=1}^{K} \alpha_k = 1$, where $\alpha_k > 0$, $k = 1, 2, \cdots, K$.

Conditional mean and variance

$$E(x_t|\mathcal{F}_{t-1}) = \sum_{k=1}^{K} \alpha_k \mu_{k,t}$$

$$\mathrm{var}(x_t|\mathcal{F}_{t-1}) = \sum_{k=1}^{K} \alpha_k \sigma_k^2 + \sum_{k=1}^{K} \alpha_k \mu_{k,t}^2 - \left(\sum_{k=1}^{K} \alpha_k \mu_{k,t} \right)^2 .$$

- $\mathrm{var}(x_t|\mathcal{F}_{t-1})$ changes with conditional means of different components.
- The shape of the conditional distributions of the time series changes with time.
- The MAR models can handle heteroscedasticity, which is common in financial time series.

Why MAR models

- Mixtures of stationary and non-stationary components can yield a stationary process.
- To handle non-stationary time series, one can just include a unit root in each component.
- Possible to capture more (or any) time series features, since different specifications of finite mixtures have been shown to be able to approximate large nonparametric classes of conditional multivariate densities (Jiang and Tanner, 1999; Li et al., 2010; Norets, 2010).

Simulation settings

Parameter	Description	Values
n	Length of time series	$U\{30, 60\}$ for yearly; 60 for quarterly; 120 for monthly data
P	Period of time series	$U\{1, 4, 12\}$
α_k	Weights of mixture components	$U(0, 1)$ for α_1; $U(0, 1 - \alpha_1)$ for α_2, and so forth
ϕ_i	Coefficients of the AR part	$N(0, 0.5)$

Projection and visualisation in 2D space

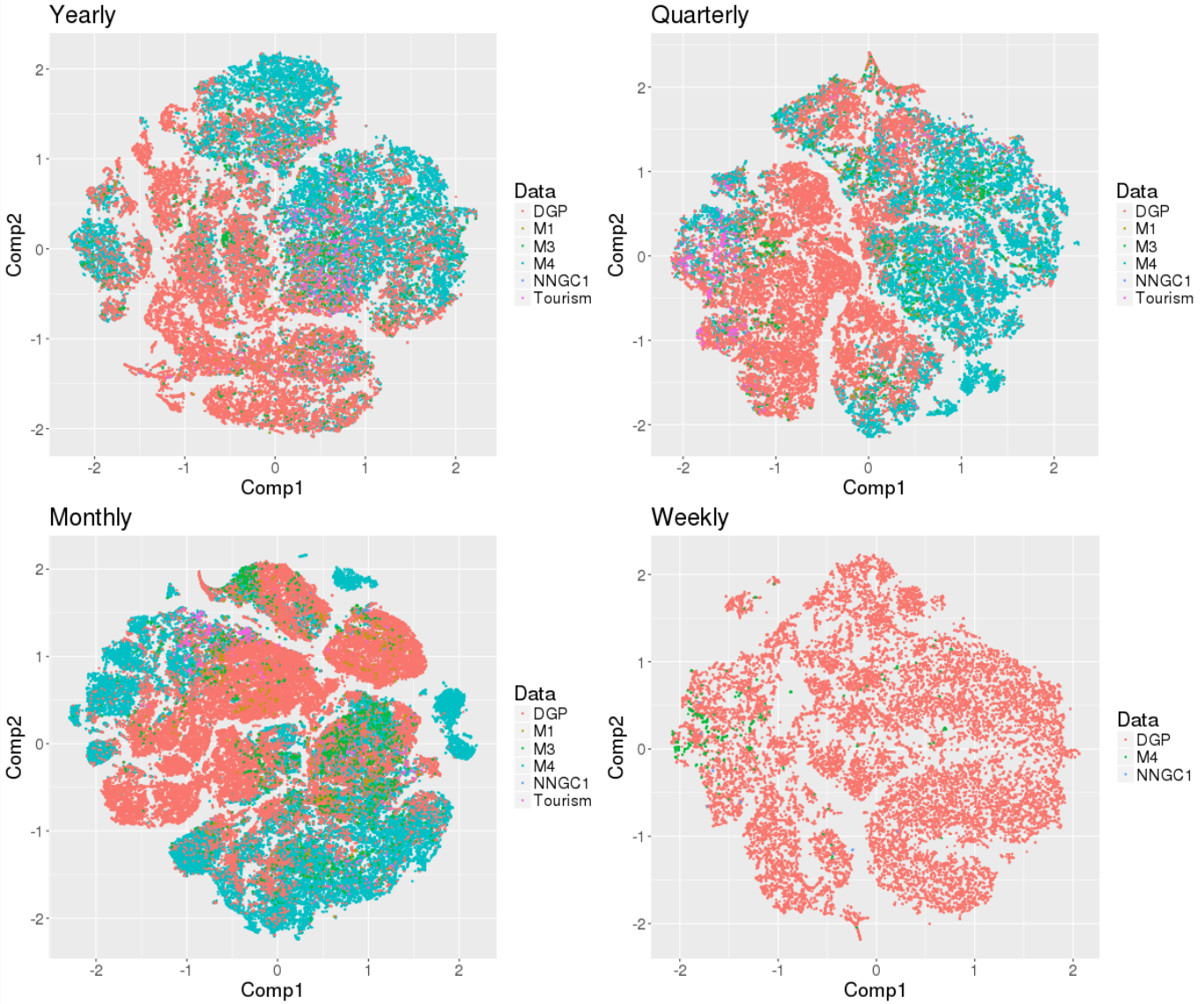
t-Stochastic Neighbor Embedding (t-SNE)

- Main idea: convert the distances to conditional probabilities and minimize the mismatch (kullback-Leibler divergence) between probabilities before and after the mapping.
- Nonlinear, and retaining both local and global structure (Maaten and Hinton, 2008; Maaten, 2014).

PCA

- Linear, and putting more emphasize on keeping dissimilar data points far apart

Investigating the coverage of MAR models



Miscoverage

We define the miscoverage of dataset A over dataset B as:

- Find the maximum ranges of the x and y axes reached by the two datasets A and B, and cut the x and y dimensions into $N_b = 30$ bins.
- In the constructed two-dimensional grid with $N_b^2 = 900$ subgrids, we denote $\mathcal{I}_{i,A} = 0$ if no points in dataset A fall into the ith subgrid. $\mathcal{I}_{i,A} = 1$ otherwise. The same defination of $\mathcal{I}_{i,B}$ applies for dataset B.
- The miscoverage of dataset A over dataset B is defined as

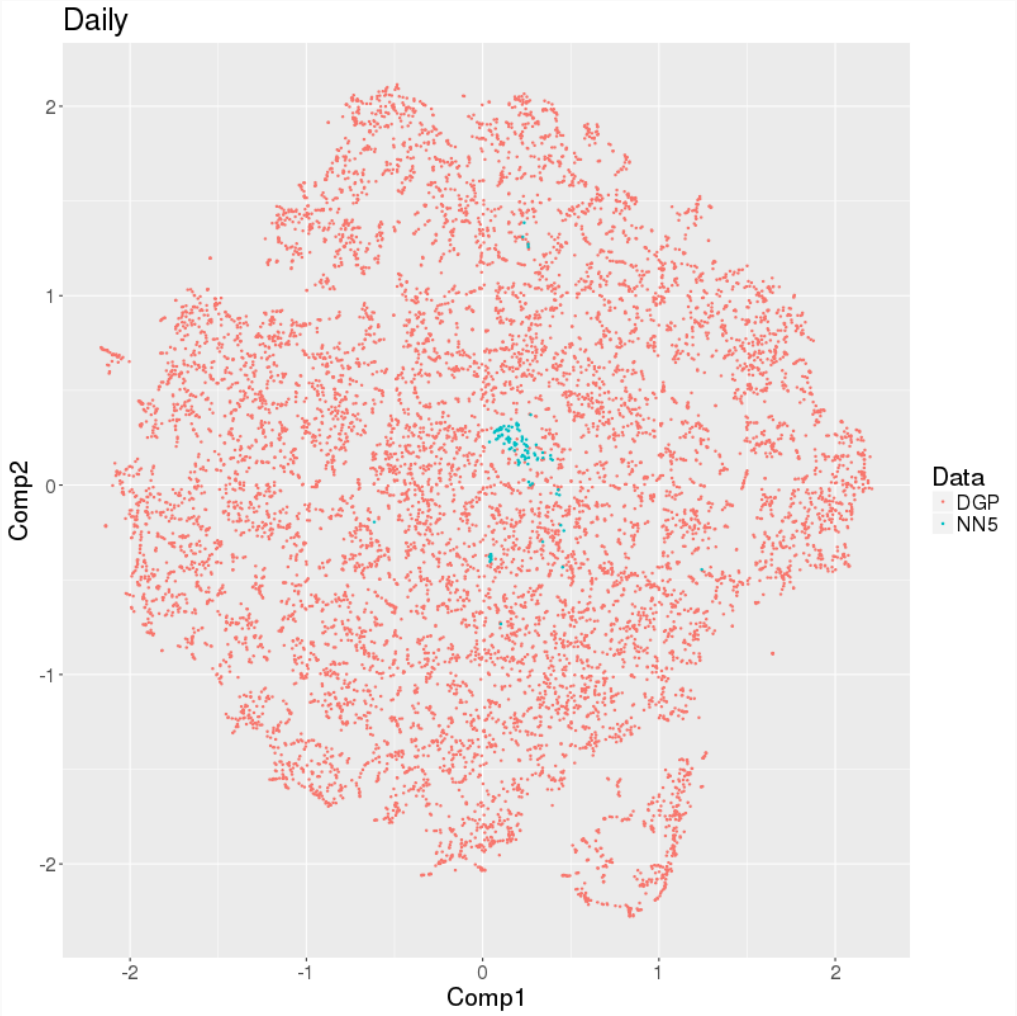
$$\text{miscoverage}_{A/B} = \frac{\sum_{i=1}^{N_b} [(1 - \mathcal{I}_{i,A}) * \mathcal{I}_{i,B}]}{N_b^2}.$$

Miscoverage

Dataset A	Dataset B					
	DGP	M4	M3	M1	Tourism	NNGC1
				Yearly		
DGP	0.00	0.02	0.01	0.00	0.00	0.00
M4	0.06	0.00	0.01	0.00	0.00	0.00
M3	0.35	0.31	0.00	0.04	0.05	0.00
M1	0.55	0.50	0.25	0.00	0.09	0.01
Tourism	0.51	0.47	0.22	0.05	0.00	0.01
NNGC1	0.66	0.61	0.34	0.13	0.20	0.00
				Quarterly		
DGP	0.00	0.04	0.01	0.00	0.00	0.00
M4	0.09	0.00	0.01	0.00	0.00	0.00
M3	0.42	0.34	0.00	0.04	0.08	0.01
M1	0.53	0.47	0.16	0.00	0.10	0.01
Tourism	0.53	0.46	0.20	0.10	0.00	0.01
NNGC1	0.65	0.58	0.26	0.13	0.14	0.00
				Monthly		
DGP	0.00	0.06	0.00	0.00	0.00	0.00
M4	0.07	0.00	0.00	0.01	0.00	0.00
M3	0.36	0.32	0.00	0.06	0.03	0.00
M1	0.45	0.42	0.16	0.00	0.06	0.00
Tourism	0.59	0.54	0.27	0.21	0.00	0.01
NNGC1	0.68	0.63	0.34	0.26	0.12	0.00
				Weekly		
DGP	0.00	0.00				0.00
M4	0.59	0.00				0.01
M3						
M1						
Tourism						
NNGC1	0.66	0.09				0.00

Extension to multiple seasonal time series

Simulation of multiple seasonal time series



New time series generation based on MAR models

New time series generation

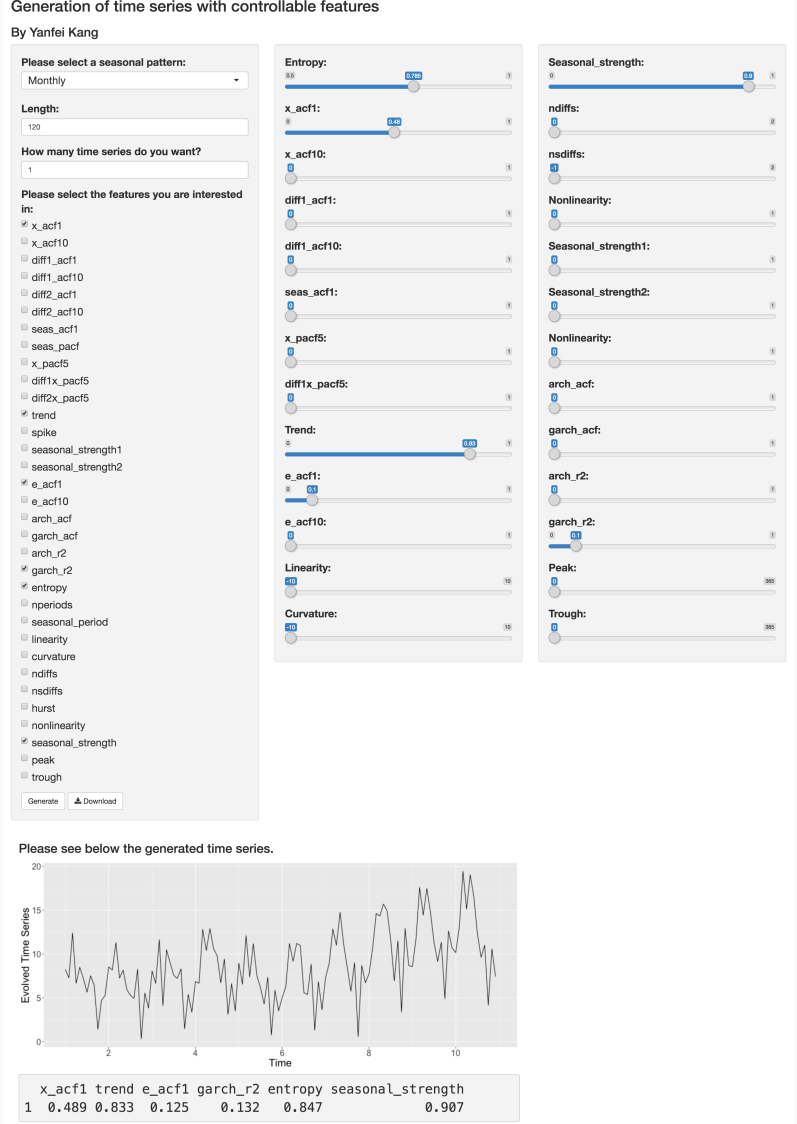
- Time series $\rightarrow$ features ✓
- Time series $\leftarrow$ features ?
  - Genetic Algorithm (GA) to evolve time series with length n
  - GA to tune the MAR model parameters $\Theta = (\alpha_k, \phi_i)$

GA procedure

- Firstly decide on the period P and length n.
- Given a target T_i in the feature space. Find Θ^* that can simulate X_{T_i} with its feature vector T_i.
- Generate an initial population of size N_P for the parameter vector Θ from the entire possible ranges.
- For each iteration, repeat the steps below.
    1. For each member in the current population, simulate a time series j and calculate its feature vector F_j.
    2. Calculate the fitness value for each member:

$$\text{Fitness}(j) = -||F_j - T_i||.$$

    3. Produce the new generation based on the crossover, mutation and the survival.
- Upon convergence, we keep the cloest time series.



- R package: `tsgeneration` available on GitHub

  https://github.com/ykang/tsgeneration

Forecasting based on features

Time series forecasting methods

The six popular forecasting methods in time series

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.

Modelling features and forecasting performances on training data

$$\mathbf{MASE_{N\times 6}} \Leftrightarrow \mathbf{F_{N\times p}}$$

$$\mathbf{MASE^{(i)}} = f_1^{(i)}(F_1) + f_2^{(i)}(F2) + ... + f_p^{(i)}(F_p) + \epsilon^{(i)}$$

- This relationship is obviously nonlinear. We use the Bayesian spline regressions to capture the nonlinearity (Li and Villani, 2013).
- R package: `movingknots` available on GitHub
  https://github.com/feng-li/movingknots

Apply the model on the forecasts on M3 (out-of-sample)

Method	Yearly		Quarterly		Monthly		All	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Naïve	3.172	2.267	1.464	1.044	1.175	0.927	1.707	1.135
Seasonal naïve	3.172	2.267	1.425	1.176	1.146	0.969	1.683	1.146
Theta	2.773	1.985	**1.114**	0.842	0.889	0.751	1.379	0.886
ETS	2.879	1.961	1.188	0.868	0.865	**0.716**	1.410	0.870
ARIMA	2.964	1.864	1.187	0.843	0.877	0.727	1.436	0.872
STL-AR	2.953	1.854	1.911	1.687	1.268	1.011	1.824	1.246
Method Selection	**2.746**	**1.782**	1.129	**0.813**	**0.855**	0.724	**1.360**	**0.857**

Conclusions

- Feature description.
- Time series simulation from MAR models.
- 2D space (identify unusual time series, find clusters, etc.).
- Develop meta-forecasting algorithms which choose a specific method based on the location of a time series in the instance space.
- Generate new time series with specific features.

Working in progress

- Extension to multivariate time series.
- Density forecasting.
- Framework on non-time series.

References

Adam, E. E. (1973), "Individual item forecasting model evaluation," *Decision Sciences*, 4, 458–470.

Fulcher, B. D. (2018), "Feature-based time-series analysis," in *Feature engineering for machine learning and data analytics*, CRC Press, pp. 87–116.

Hyndman, R. J., Wang, E., and Laptev, N. (2015), "Large-scale unusual time series detection," in *Proceedings of the IEEE International Conference on Data Mining*, Atlantic City, NJ, USA, 14–17 November 2015.

Jiang, W. and Tanner, M. A. (1999), "On the approximation rate of hierarchical mixtures-of-experts for generalized linear models," *Neural Computation*, 11, 1183–1198.

Kang, Y., Hyndman, R. J., and Smith-Miles, K. (2017), "Visualising forecasting algorithm performance using time series instance spaces," *International Journal of Forecasting*, 33, 345–358.

Li, F. and Villani, M. (2013), "Efficient bayesian multivariate surface regression." *Scandinavian Journal of Statistics*, 40, 706–723.

Li, F., Villani, M., and Kohn, R. (2010), "Flexible modeling of conditional distributions using smooth mixtures of asymmetric student-*t densities," Journal of Statistical Planning and Inference, 140, 3638–3654.*

Maaten, L. v. d. (2014), "Accelerating t-SNE using tree-based algorithms," The Journal of Machine Learning Research, 15, 3221–3245.

Maaten, L. v. d. and Hinton, G. (2008), "Visualizing data using t-SNE," Journal of Machine Learning Research, 9, 2579–2605.

Makridakis, S. and Hibon, M. (2000), "The M3-Competition: results, conclusions and implications," International Journal of Forecasting, 16, 451–476.

Meade, N. (2000), "Evidence for the selection of forecasting methods," Journal of Forecasting, 19, 515–535.

Norets, A. (2010), "Approximation of conditional densities by smooth mixtures of regressions," Annals of Statistics, 38, 1733–1766.

Petropoulos, F., Makridakis, S., Assimakopoulos, V., and Nikolopoulos, K. (2014), "'Horses for Courses' in demand forecasting," European Journal of Operational Research, 237, 152–163.

Wolpert, D. H. (1996), "The lack of a priori distinctions between learning algorithms," Neural computation, 8, 1341–1390.

Wong, C. S. and Li, W. K. (2000), "On a mixture autoregressive model," Journal of the Royal Statistical Society: Series B (Statistical Methodology), 62, 95–115.

Thanks!

http://feng.li/

feng.li@cufe.edu.cn