Time series forecasting based on automatic feature extraction

Feng Li

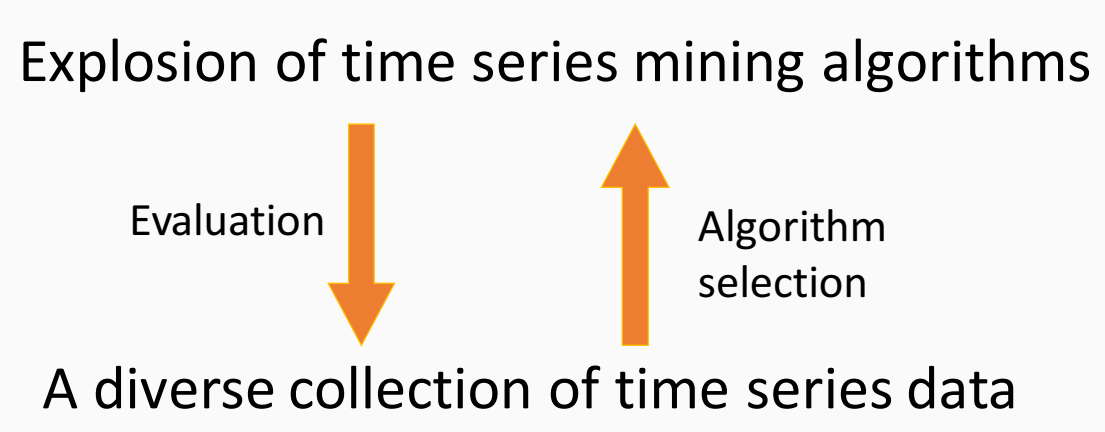
School of Statistics and Mathematics
Central University of Finance and Economics

- Joint with Yanfei Kang and Xixi Li.
- Feng Li's research is supported by National Natural Science Foundation of China.

Outline

Motivation

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



- Is there an efficient way to **forecast which algorithm works the best** for a particular time series?

Literature

- Features of time series → benefits in producing more accurate forecasting accuracies (Adam 1973).
- Features → forecasting method selection rules (Meade 2000).
- "Horses for courses" → 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 → to get better understanding of their relative performances (Kang et al. 2017).

Traditional time series features

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?

- **Bad News**: 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.

Encoding time series to images

- Let $R(i,j)$ be the element of the **time series image matrix** where i indexes time on the x-axis of the recurrence plot and j indexes time on the y-axis;
- **Recurrence Plot Encoding**

$$R(i,j) = \begin{cases} S, & \textit{if } \| \overrightarrow{x(i)} - \overrightarrow{x(j)} \| / \epsilon > S, \\ \| \overrightarrow{x(i)} - \overrightarrow{x(j)} \| / \epsilon, & \textit{otherwise}, \end{cases}$$

  where S is the threshold distance and ϵ is some small number.
- **Gramian Angular Field Encoding** (Wang & Oates 2015): Given a time series $X = x_1, x_2, ..., x_n$, we scale the series X into $[-1,\ 1]$.

$$x_i = \frac{(x_i - \max(X) + (x_i - \min(X))}{\max(X) - \min(X)}$$

  Then, we convert the scaled time series X into "polar coordinates".

Encoding time series to images

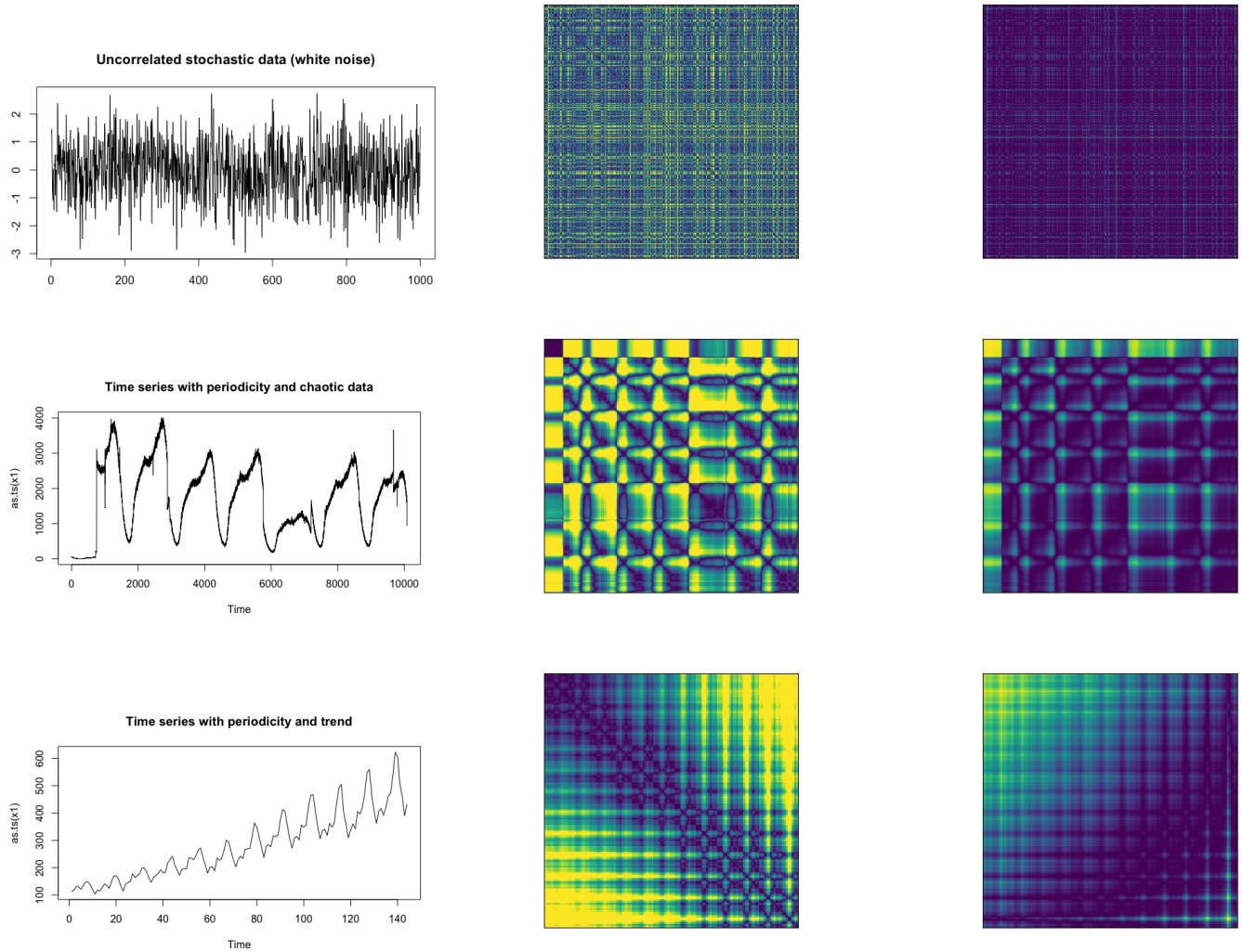


Figure: Typical examples of recurrence plot(the second column) and Gramian Angular Field(the third column)

Image features

- The original Bag of Features (BoF) model, which extracts features from one-dimensional signal segments, has achieved a great success in time series classification (Baydogan et al. 2013, Wang et al. 2013).
- Hatami et al. (2017) transform time-series into two-dimensional recurrence images with recurrence plot (Eckmann et al. 1987) and then applies the BoF model.
- (Razavian et al. 2014) use the features acquired by the convolutional neural network as the input of the classifier, which significantly improves the accuracy of image classification.

Image-based time series feature extraction
↛ Scale-invariant feature transform (SIFT)

- The scale space of an image is defined as the original image is convoluted with a variable-scale 2-dimensional Gaussian function.
- Key points are then taken as maxima/minima of the difference of Gaussians that occur at multiple scales.
- In our study, we use a 128-elements vector to characterize the key descriptors.
- Firstly, we establish an 8-direction histogram in each 4×4 sub-region, and a total of 16 sub-regions in the 16×16 region around the key point are calculated. Then we calculate the magnitude and direction of each pixel's gradient magnitude and add to the sub-region.
- In the end, a total of 128-dimensional image data based on histograms are generated.
- The sift method calculates the distribution characteristics of feature points in the whole image, and then generates a global histogram, so the spatial distribution information of the image is lost, and the image may not be accurately identified.
- A spatial pyramid method statistically distributes image feature points at different resolutions to obtain spatial information of images.

Image-based time series feature extraction
↛ Scale-invariant feature transform (SIFT)

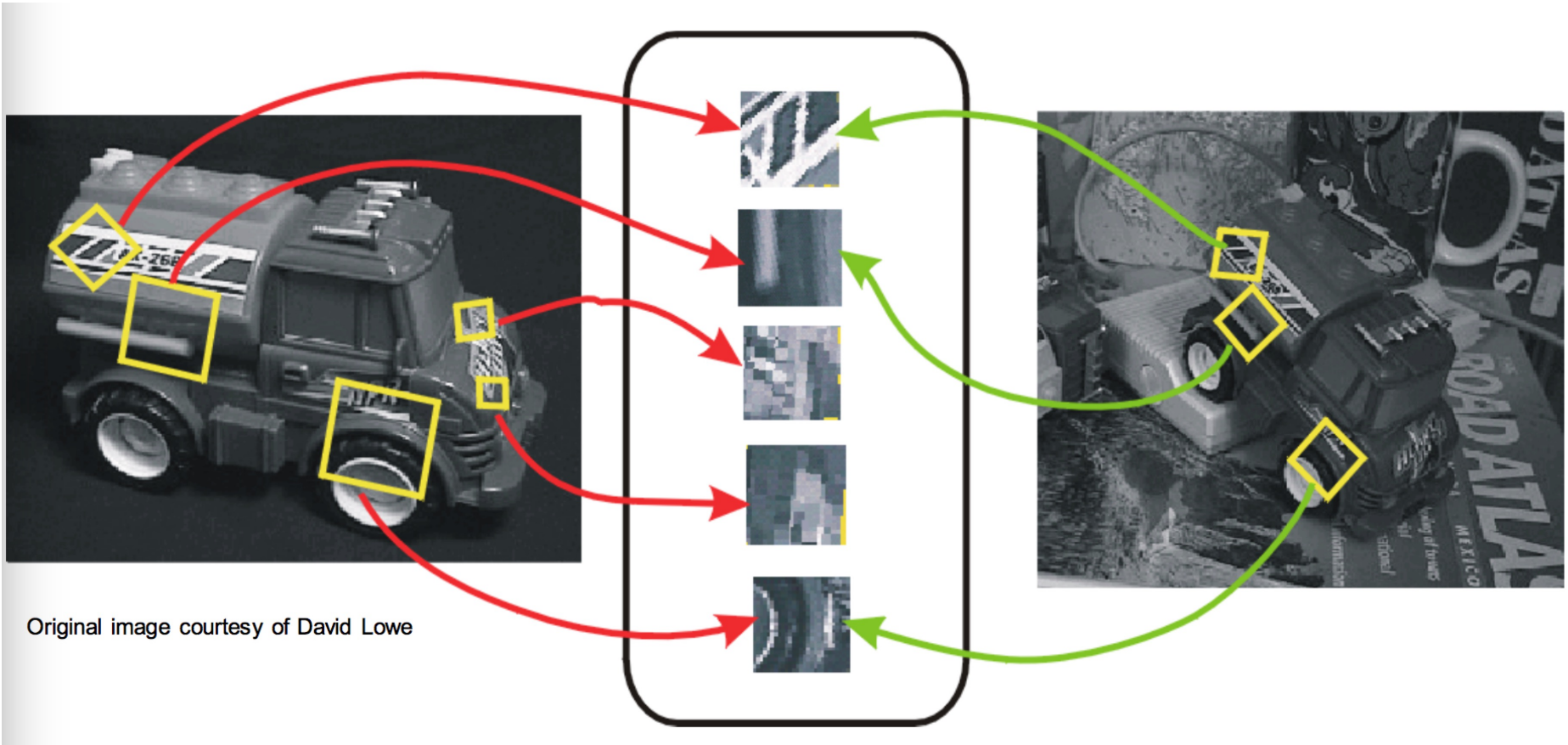


Figure: Image feature extraction with scale-invariant feature transform

Image-based time series feature extraction

↛ Scale-invariant feature transform (SIFT)

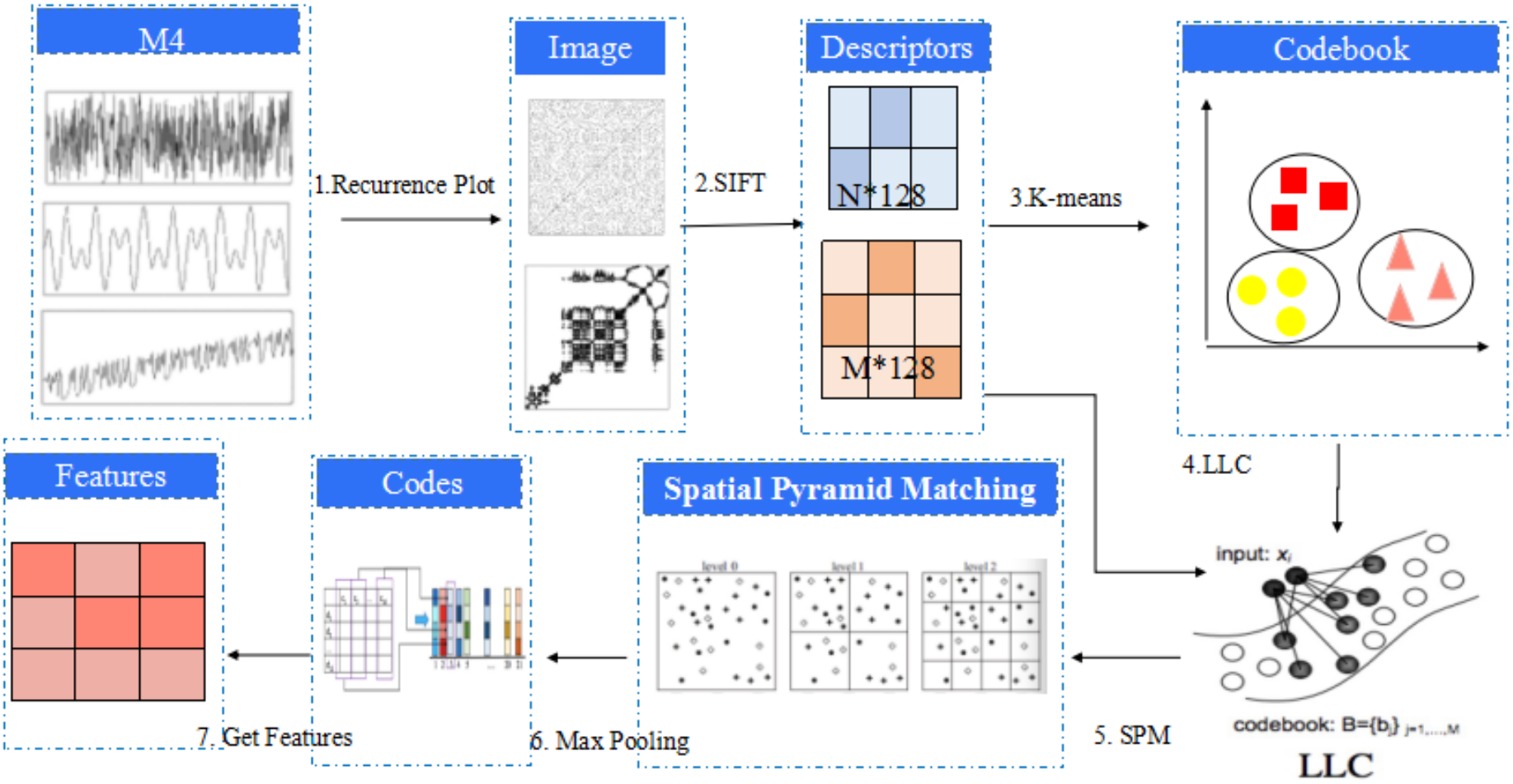


Figure: Image Feature Extraction with scale-invariant feature transform

Image-based time series feature extraction
↛ Transfer learning with fine-tuning

- The deep convolutional neural networks has greater advantages in accuracy compared with the traditional image features.
- But building models from scratch is complex and time consuming.
- One could used a pretrained trained neural network model and make adjustments to her own task – **Transfer learning**.

Image-based time series feature extraction

↛ Transfer learning with fine-tuning

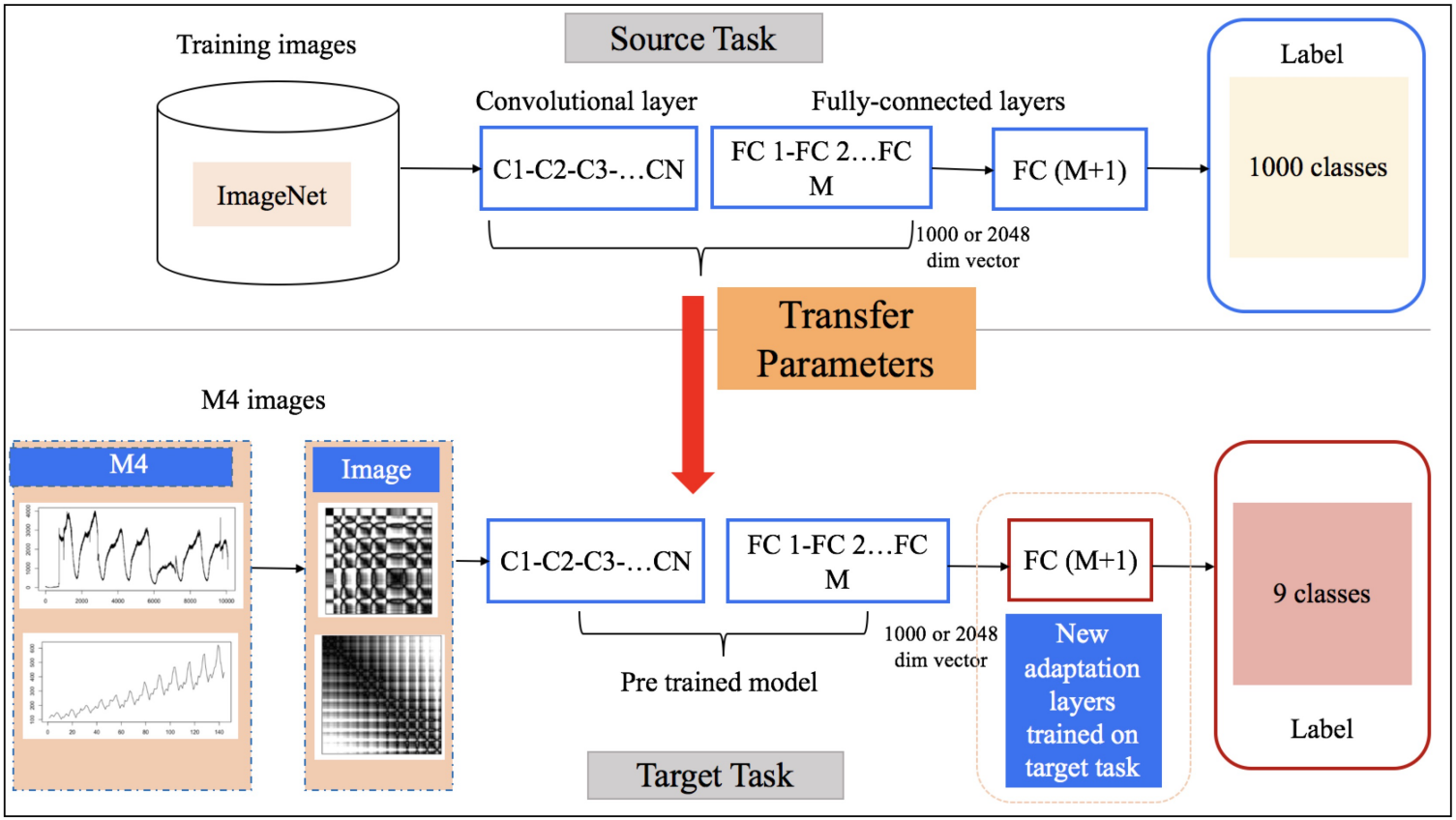


Figure: Transfer Learning-Fine-Tuning. We can train the Inception(Szegedy, Vanhoucke, Ioffe, Shlens & Wojna 2016), ResNet(Szegedy, Ioffe & Vanhoucke 2016), VGG(Simonyan & Zisserman 2014) and other classic CNN models with large dataset ImageNet(Deng et al. 2009). With the pre trained model, we fix the parameters of the previous layers, and fine-tune the next few layers for our task. In this way, the speed of network training will be greatly accelerated, and it will also greatly promote the performance of our task.

Time series forecasting methods

- **Naive**: uses the most recent observation as the forecast for all future periods.
- **Seasonal naive**: forecasts are equal to the most recent observation from the corresponding time of year.
- **The Theta method**: It performed particularly well in the M3-Competition proposed by Assimakopoulos & Nikolopoulos (2000).
- **ETS**: exponential smoothing state space modeling, which is used widely as a general forecasting algorithm for trended and seasonal time series proposed by Hyndman et al. (2017).
- **ARIMA**: autoregressive integrated moving average models, as implemented in the automated algorithm by Hyndman & Khandakar (2008).
- **STL-AR**: an AR model is fitted to the seasonally adjusted series obtained from a STL decomposition proposed by Cleveland et al. (1990).
- **Nnetar**: It fits a neural network model to a time series with lagged values of the time series as inputs (and possibly some other exogenous inputs).
- **Rw-drift**: Random Walk with Drift.
- **Tbats**: A Tbats model differs from dynamic harmonic regression in that the seasonality is allowed to change slowly over time in a Tbats model, while harmonic regression terms force the seasonal patterns to repeat periodically without changing, as implemented in the automated algorithm by Hyndman & Khandakar (2008).

Forecast loss measurement

- **Forecast loss measurement**: Overall Weighted Average (OWA) is an indicator of two accuracy measures: the Mean Absolute Scaled Error (MASE) and the symmetric Mean Absolute Percentage (sMAPE).

$$\mathrm{sMAPE} = \frac{1}{h}\sum_{t=1}^{h}\frac{2 \mid Y_t - \widehat{Y}_t \mid}{\mid Y_t \mid + \mid \widehat{Y}_t \mid},$$

$$\mathrm{MASE} = \frac{1}{h}\frac{\sum_{t=1}^{h} \mid Y_t - \widehat{Y}_t \mid}{\frac{1}{n-m}\sum_{t=m+1}^{n} \mid Y_t - Y_{t-m} \mid},$$

$$\mathrm{OWA} = \frac{\mathrm{sMAPE/sMPAE} + \mathrm{MASE/MASE}}{2},$$

- Train a high dimensional regression model (Lasso) with X_{train} and $MASE$
- Calculate predicted $MASE$ with X_{test} using R_i

Model selection

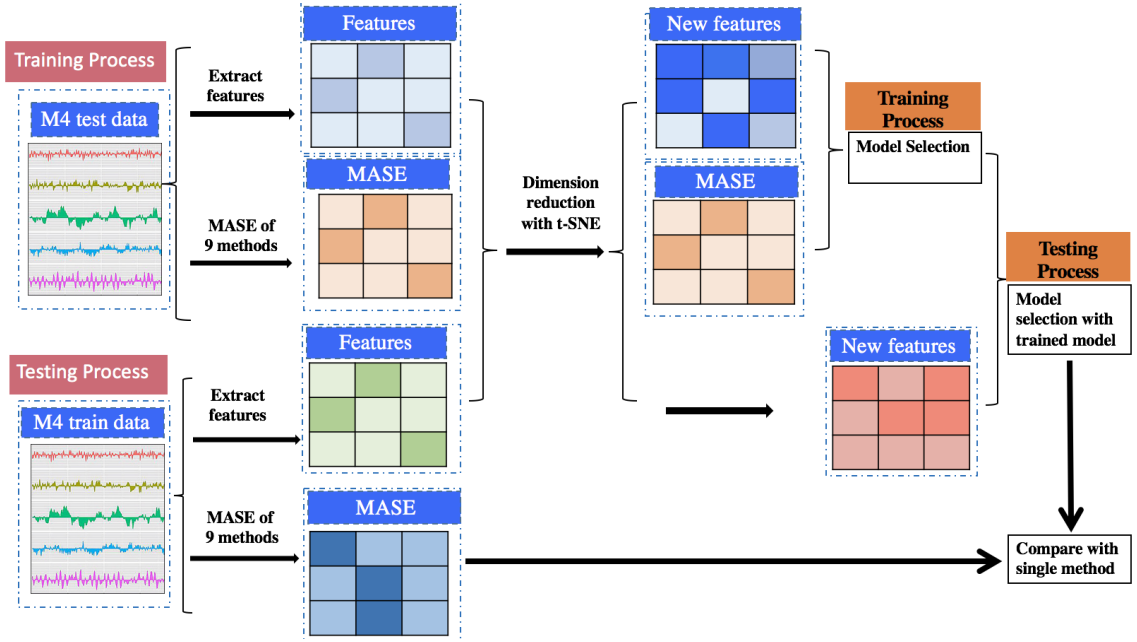


Figure: Model selection framework for the largest time series forecasting competition dataset - M4 (Makridakis et al. 2018) based on automatic features.

Model selection

Forecasting Method	Yearly	Quarterly	Monthly	Weekly	Daily	Hourly	Total
			Single method				
auto_arima	3.45	1.17	0.93	2.38	3.35	0.94	1.67
ets	3.44	1.16	0.95	2.53	3.25	1.82	1.68
nnetar	4.05	1.55	1.15	3.84	4.13	1.07	2.05
tbats	3.44	1.19	1.05	2.49	3.28	1.23	1.73
stlm_ar	10.37	2.03	1.33	39.67	31.2	1.49	4.98
rw_drift	3.07	1.33	1.18	2.68	3.25	11.46	1.79
thetaf	3.37	1.23	0.97	2.64	3.26	2.45	1.69
naive	3.97	1.48	1.21	2.78	3.28	11.61	2.04
snaive	3.97	1.6	1.26	2.78	3.28	1.19	2.06
Min	3.07	1.16	0.93	2.38	3.25	0.94	1.67
			Model selection+Recurrence plot				
SIFT + *Lasso*	3.45	1.18	**0.93**	**2.38**	3.36	**0.94**	1.68
SIFT + *SVM* + *rbf*(10)	3.42	1.36	1.00	7.81	6.61	**0.84**	1.90
SIFT + *SVM* + *rbf*	3.45	1.17	**0.93**	**2.38**	3.35	**0.94**	**1.67**
Pre trained CNN model+Classifier							
inception − *v*1 + *Lasso*	3.45	1.18	**0.93**	**2.37**	3.35	**0.94**	**1.67**
resnet − *v*1 − 101 + *Lasso*	3.45	1.17	**0.93**	2.38	3.35	**0.94**	**1.67**
resnet − *v*1 − 50 + *Lasso*	3.45	1.17	**0.93**	2.38	3.35	**0.94**	**1.67**
vgg − 19 + *Lasso*	3.45	1.18	**0.93**	2.70	3.52	**0.94**	1.69
			Model selection+Gramian angular field				
Pre trained CNN model+Classifier							
inception − *v*1 + *Lasso*	3.47	1.18	0.94	**2.38**	3.38	**0.94**	1.69
resnet − *v*1 − 101 + *Lasso*	3.45	1.18	**0.93**	**2.37**	3.35	**0.93**	1.68
resnet − *v*1 − 50 + *Lasso*	3.45	1.18	**0.93**	**2.35**	3.35	**0.93**	1.68
vgg − 19 + *Lasso*	3.45	1.17	0.95	2.53	3.34	1.81	1.69

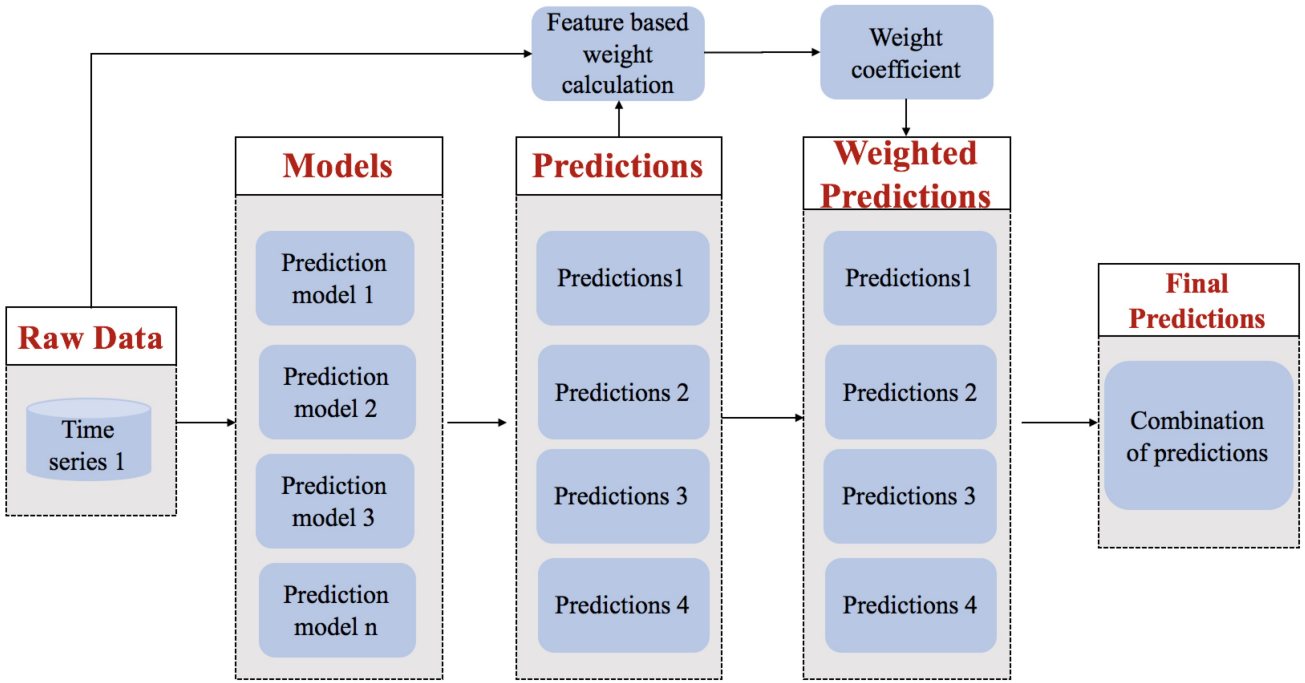
Model averaging

- In order to get the weight $w(f_n)_m$ for every method, softmax transform is carried on the output $p(f_n)_m$.

$$w(f_n)_m = \frac{e^{p(f_n)_m}}{\sum_m e^{p(f_n)_m}}$$

- The weighted average loss function is minimized:

$$\text{argmin}_w \overline{L_n} = \sum_{m=1}^{M} w(f_n)_m O_{nm}$$



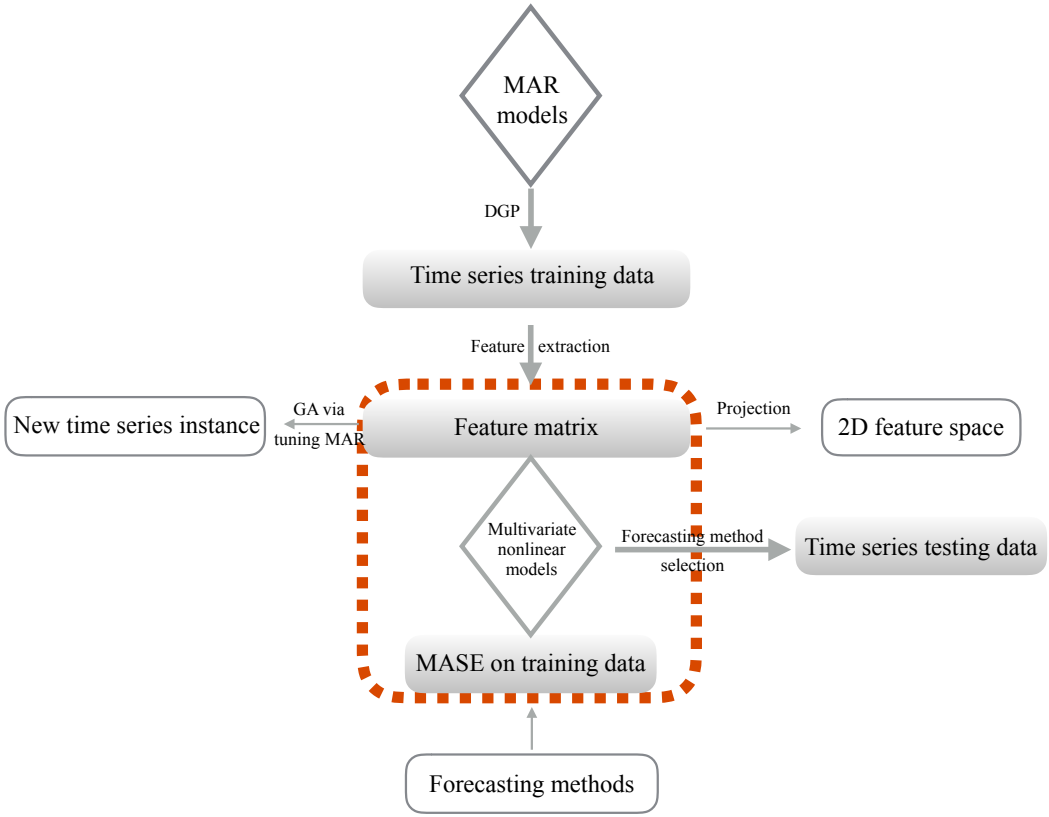
Model averaging

rank	Yearly	Quarterly	Monthly	Weekly	Daily	Hourly	Total
			M4 competition				
1	0.78	0.85	0.84	0.85	1.05	0.44	0.82
2	0.80	0.85	0.86	0.80	1.02	0.48	0.84
3	0.82	0.86	0.87	0.77	0.81	0.44	0.84
4	0.81	0.86	0.85	0.80	1.00	0.47	0.84
5	0.80	0.85	0.87	0.90	0.98	0.67	0.84
6	0.81	0.85	0.88	0.75	0.98	0.66	0.85
7	0.80	0.91	0.88	0.96	1.06	0.65	0.86
8	0.79	0.90	0.90	0.97	1.00	1.01	0.86
9	0.84	0.88	0.88	0.78	1.00	0.41	0.86
10	0.82	0.88	0.90	0.94	0.99	0.48	0.87
			Model averaging+Recurrence plot				
SIFT + *XGBoost*	0.82	0.85	0.89	0.92	1.04	0.50	0.86
Pre trained CNN model+Classifier							
inception − *v*1 + *XGBoost*	0.82	0.86	0.88	0.88	1.02	0.51	0.85
resnet − *v*1 − 101 + *XGBoost*	0.82	0.85	0.87	0.88	1.02	0.50	0.85
resnet − *v*1 − 50 + *XGBoost*	0.82	0.86	0.88	0.87	1.02	0.50	0.85
vgg − 19 + *XGBoost*	0.82	0.86	0.88	0.88	1.01	0.50	0.85
			Model averaging+Gramian angular field				
Pre trained CNN model+Classifier							
inception − *v*1 + *Lasso*	0.82	0.86	0.87	0.86	1.02	0.51	0.85
resnet − *v*1 − 101 + *Lasso*	0.82	0.86	0.88	0.83	1.03	0.50	0.85
resnet − *v*1 − 50 + *Lasso*	0.82	0.85	0.89	0.84	1.02	0.50	0.85
vgg − 19 + *Lasso*	0.82	0.86	0.88	0.84	1.02	0.51	0.86

Automatic time series forecasting

- Automatic time series features are extracted via machine learning (deep learning) algorithms.
- **The advantage:** The automatic extracted features are usually **not interpretable**.
- This allows for statistical forecasting when data privacy is a real concern.

Automatic time series forecasting



- Working papers for the *automatic time series forecasting* framework
  - 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)

Automatic time series forecasting
↛ Automatic generated time-series-Net

- Is ImageNet a good training source for time series image?
- Consist of multiple stationary or non-stationary autoregressive components.
- A K-component MAR model is defined as (Wong & 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 $\alpha_k > 0$, $k = 1, 2, \cdots, K$.

Automatic time series forecasting
↛ Automatic generated time-series-Net

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

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

- $\text{var}(x_t|\mathcal{F}_{t-1})$ 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.

Automatic time series forecasting
↛ Automatic generated time-series-Net

- 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 & Tanner 1999, Li et al. 2010, Norets 2010).

Automatic time series forecasting

↛ Automatic generated time-series-Net

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)

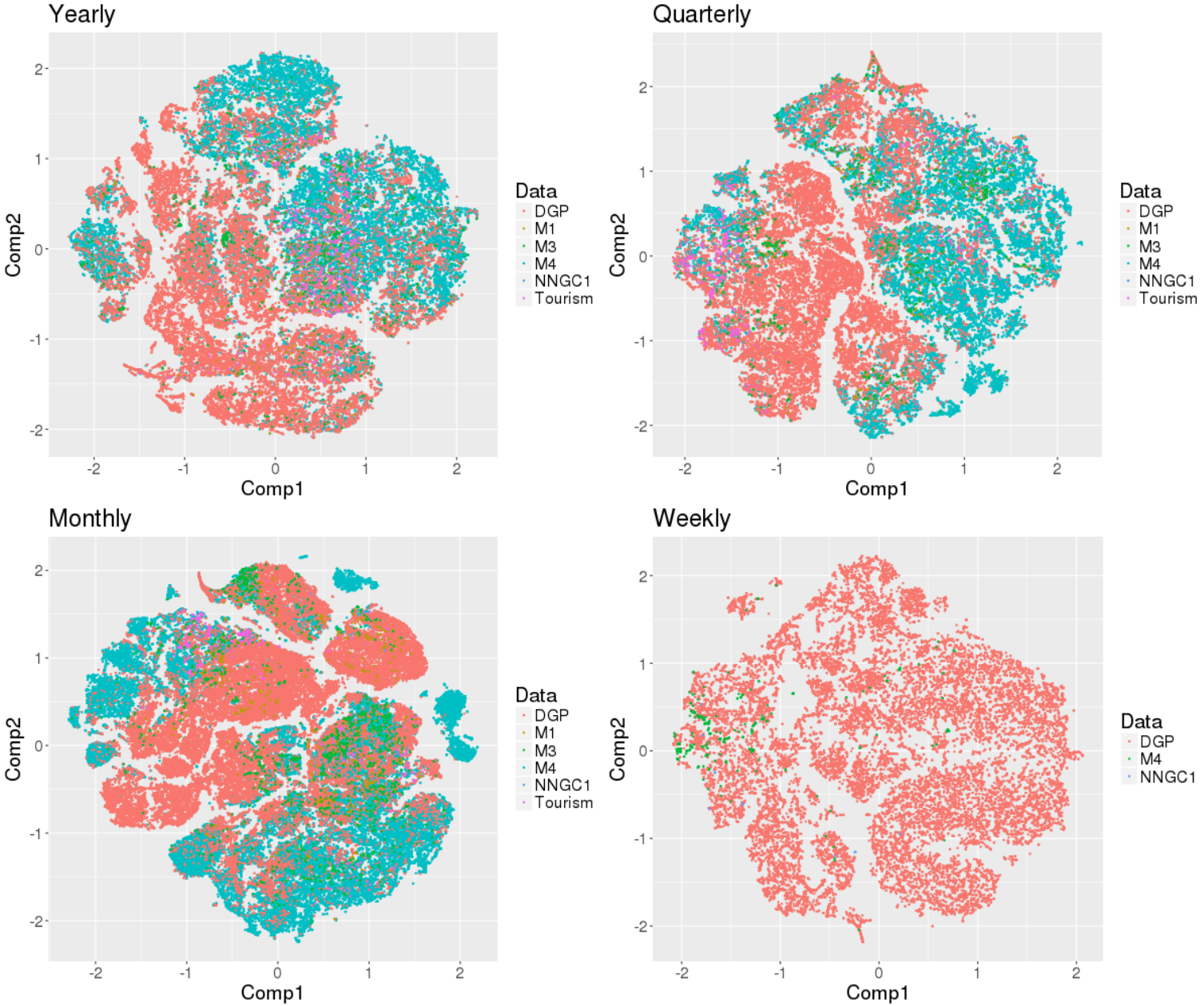
- Main idea: convert the distances to conditional probabilities and minimize the mismatch (kullback-Leibler divergence) between probabilities before and after the mapping.
- Nonlinear, and retaining both local and global structure (Maaten & Hinton 2008, Maaten 2014).

PCA

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

Automatic time series forecasting

↛ Automatic generated time-series-Net



Automatic time series forecasting
↛ Automatic generated time-series-Net

- 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

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

Automatic time series forecasting

↛ Automatic generated time-series-Net

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

Ongoing work

- Automatic density forecasting
- Multivariate time series forecasting
- Automatic forecasting for dependent time series.

Thank you!

feng.li@cufe.edu.cn

http://feng.li/

References I

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

Assimakopoulos, V. & Nikolopoulos, K. (2000), 'The theta model: a decomposition approach to forecasting', *International Journal of Forecasting* **16**(4), 521–530.

Baydogan, M. G., Runger, G. & Tuv, E. (2013), 'A bag-of-features framework to classify time series', *IEEE transactions on pattern analysis and machine intelligence* **35**(11), 2796–2802.

Cleveland, R. B., Cleveland, W. S., McRae, J. E. & Terpenning, I. (1990), 'STL: A seasonal-trend decomposition procedure based on loess', *Journal of Official Statistics* **6**(1), 3–73.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K. & Li, F. F. (2009), Imagenet: A large-scale hierarchical image database, *in* 'IEEE Conference on Computer Vision and Pattern Recognition'.

Eckmann, J.-P., Kamphorst, S. O. & Ruelle, D. (1987), 'Recurrence plots of dynamical systems', *EPL (Europhysics Letters)* **4**(9), 973.

References II

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

Hatami, N., Gavet, Y. & Debayle, J. (2017), 'Bag of recurrence patterns representation for time-series classification', *Pattern Analysis and Applications* pp. 1–11.

Hyndman, R. J. & Khandakar, Y. (2008), 'Automatic time series forecasting: the forecast package for R', *Journal of Statistical Software* **26**(3), 1–22.

Hyndman, R. J., Koehler, A. B., Snyder, R. D. & Grose, S. (2017), 'A state space framework for automatic forecasting using exponential smoothing methods', *International Journal of Forecasting* **18**(3), 439–454.

Hyndman, R. J., Wang, E. & 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. & Tanner, M. A. (1999), 'On the approximation rate of hierarchical mixtures-of-experts for generalized linear models', *Neural Computation* **11**(5), 1183–1198.

References III

Kang, Y., Hyndman, R. J. & Smith-Miles, K. (2017), 'Visualising forecasting algorithm performance using time series instance spaces', *International Journal of Forecasting* **33**(2), 345–358.

Li, F., Villani, M. & Kohn, R. (2010), 'Flexible modeling of conditional distributions using smooth mixtures of asymmetric student-*t densities*', Journal of Statistical Planning and Inference **140***(12), 3638–3654.*

Maaten, L. v. d. (2014), 'Accelerating t-SNE using tree-based algorithms', The Journal of Machine Learning Research **15***(1), 3221–3245.*

Maaten, L. v. d. & Hinton, G. (2008), 'Visualizing data using t-SNE', Journal of Machine Learning Research **9***(Nov), 2579–2605.*

Makridakis, S., Spiliotis, E. & Assimakopoulos, V. (2018), 'The m4 competition: Results, findings, conclusion and way forward', International Journal of Forecasting .

Meade, N. (2000), 'Evidence for the selection of forecasting methods', Journal of Forecasting **19***(6), 515–535.*

Norets, A. (2010), 'Approximation of conditional densities by smooth mixtures of regressions', Annals of Statistics **38***(3), 1733–1766.*

References IV

Petropoulos, F., Makridakis, S., Assimakopoulos, V. & Nikolopoulos, K. (2014), "'Horses for courses' in demand forecasting', European Journal of Operational Research **237***(1), 152–163.*

Razavian, A. S., Azizpour, H., Sullivan, J. & Carlsson, S. (2014), 'Cnn features off-the-shelf: An astounding baseline for recognition'.

Simonyan, K. & Zisserman, A. (2014), 'Very deep convolutional networks for large-scale image recognition', Computer Science .

Szegedy, C., Ioffe, S. & Vanhoucke, V. (2016), 'Inception-v4, inception-resnet and the impact of residual connections on learning'.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016), Rethinking the inception architecture for computer vision, in *'Computer Vision and Pattern Recognition'.*

Wang, J., Liu, P., She, M. F., Nahavandi, S. & Kouzani, A. (2013), 'Bag-of-words representation for biomedical time series classification', Biomedical Signal Processing and Control **8***(6), 634–644.*

Wang, Z. & Oates, T. (2015), Imaging time-series to improve classification and imputation, in *'Proceedings of the 24th International Conference on Artificial Intelligence', AAAI Press, pp. 3939–3945.*

References V

Wong, C. S. & Li, W. K. (2000), 'On a mixture autoregressive model', Journal of the Royal Statistical Society: Series B (Statistical Methodology) **62***(1), 95–115.*