Machine learning for time series: from forecasting to causal inference

TRAIL doctoral school, BeCentral

Gianluca Bontempi
Machine Learning Group
ULB, Université Libre de Bruxelles
mlg.ulb.ac.be

According to you...

.. what are the (three) most important concepts (not buzzwords :-) in supervised machine learning?

The pillars of supervised learning

1. **Dependency**: a **target y** is dependent on an **input x** if **x** brings information about **y** (i.e. knowing **x** reduces the uncertainty of **y**)

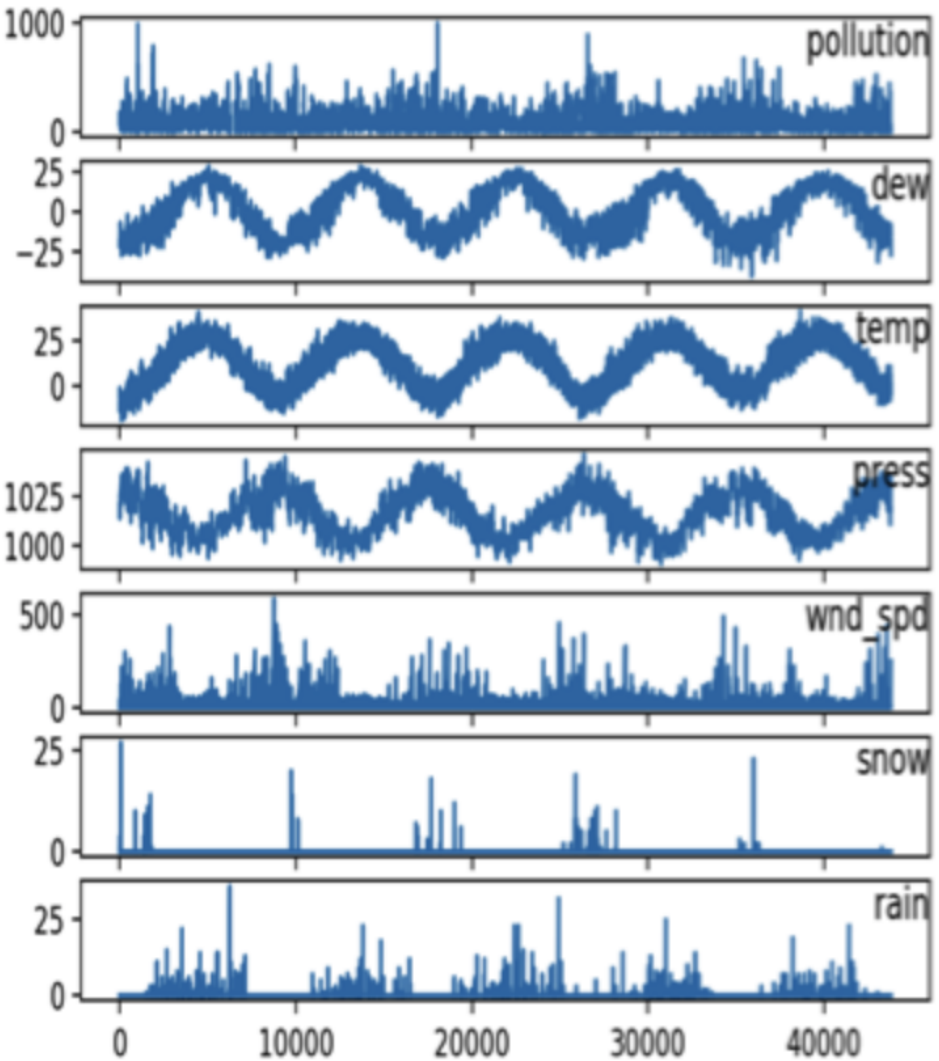
$$\mathbf{y} = f(x) + \mathbf{w}$$

2. **Estimation**: learners are estimators submitted to the bias/variance trade-off (i.e. not necessarily the most complex learner that generalises the best)
3. **Causality implies dependency but not the other way round**: "what if I observe" is not "what if I intervene"...

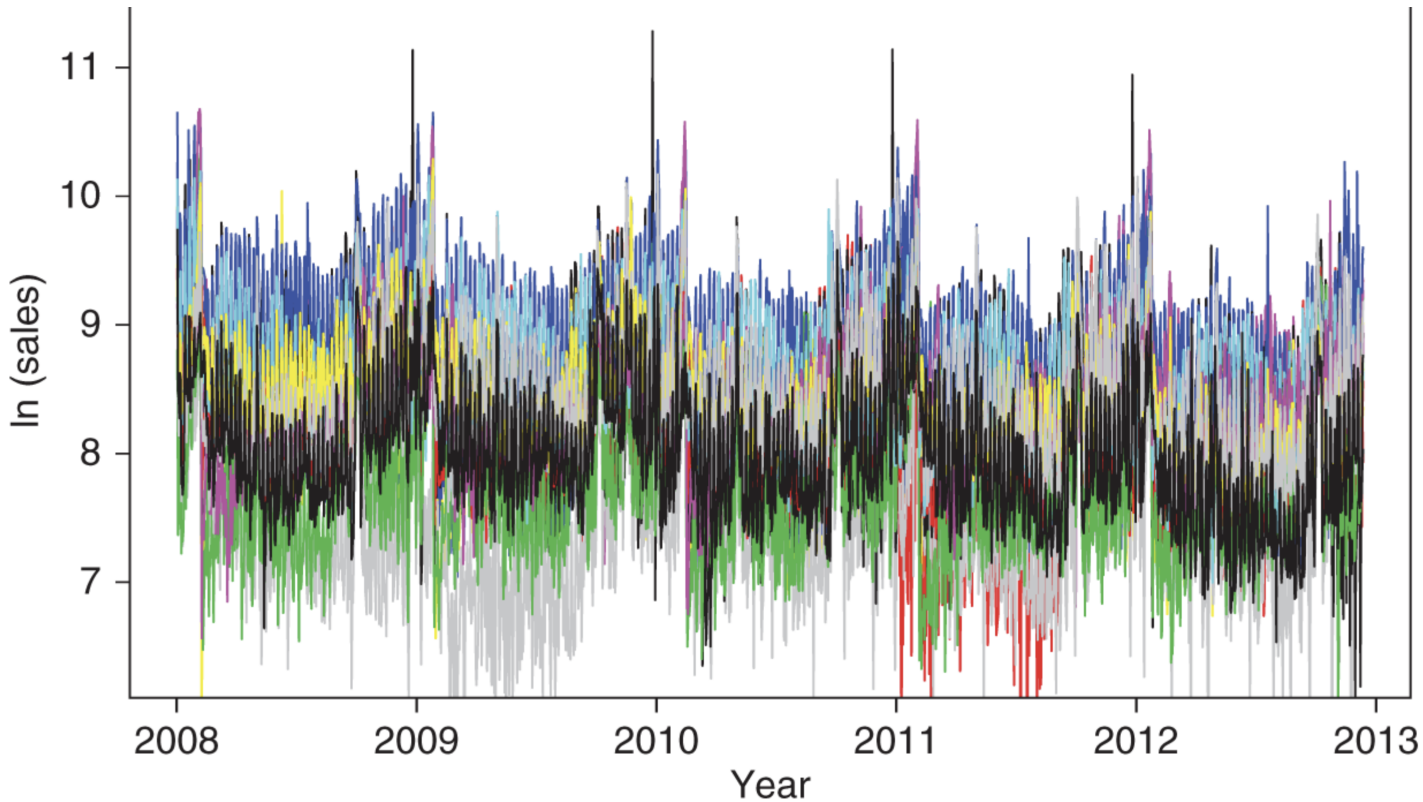
Outline

- Dependency and large-variate temporal settings (e.g. in spatio-temporal time series)
- ML and forecasting:
  - One-step-ahead univariate forecasting
  - Multi-step-ahead univariate forecasting
  - Multivariate multi-step-ahead forecasting
  - Dynamic factor model based on machine learning (DFML)
- ML and causal inference: from associative dependencies to causal relationships in multivariate temporal data.

Multivariate series: environment



Multivariate series: business



Time plots of daily sales in natural logarithms of a clothing brand in 25 provinces in China from 1 January 2008 to 9 December 2012 (from Pena and Tsay book).

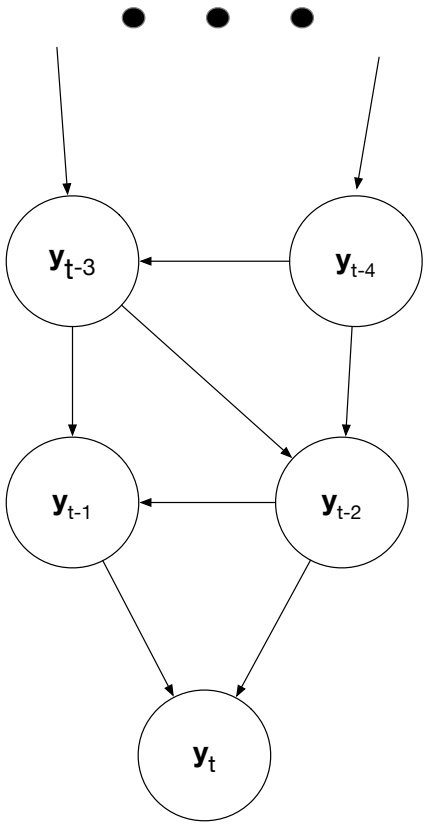
Multivariate multi-step-ahead forecasting

Probably the most difficult prediction task in the world....

- Large dimensionality
- Long prediction horizons
- Nonlinearity
- Noise
- Cross-sectional and temporal dependencies
- Nonstationarity, seasonality
- Relevant application domains: Internet of Things, finance, spatio-temporal tasks (climate, energy)

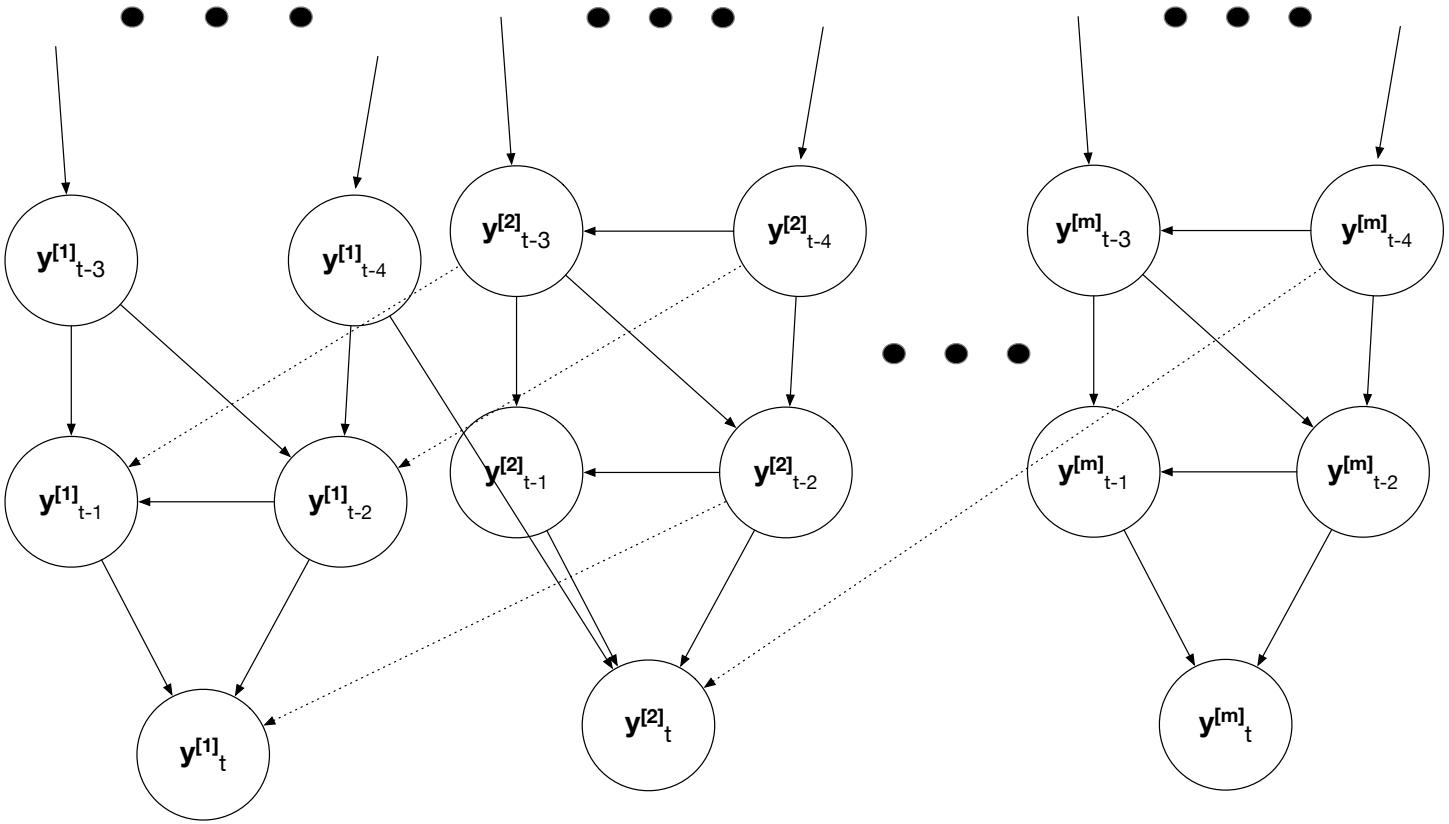
Graphical representation of dependency (univariate)

Autoregressive process $y_{t+1} = f(y_t, y_{t-1}) + w_{t+1}$



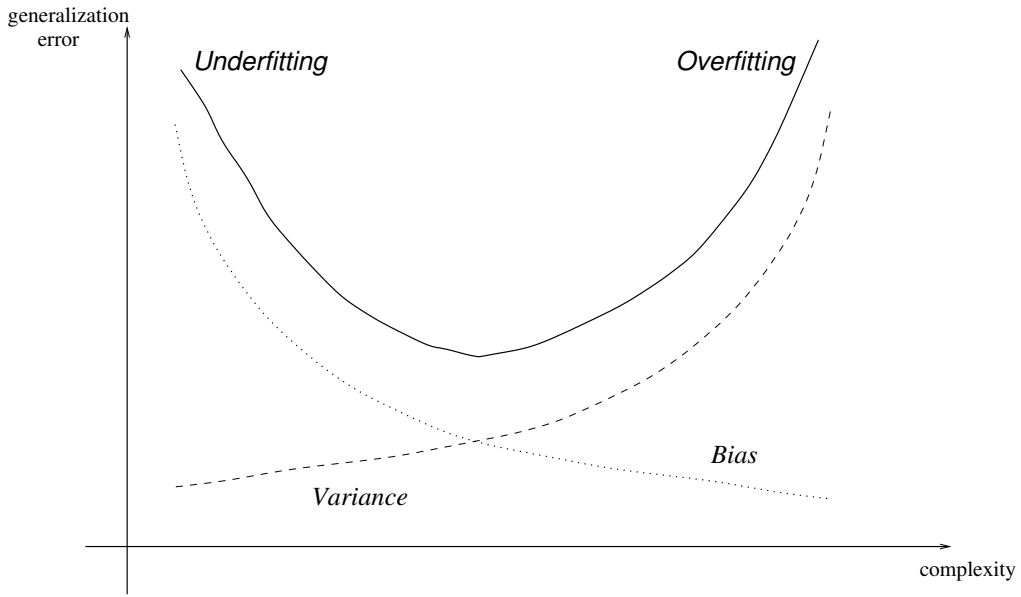
D-separation reads off the graph all existing (conditional) dependencies.

Graphical representation of dependency (multivariate)



Temporal and cross-variate dependencies.

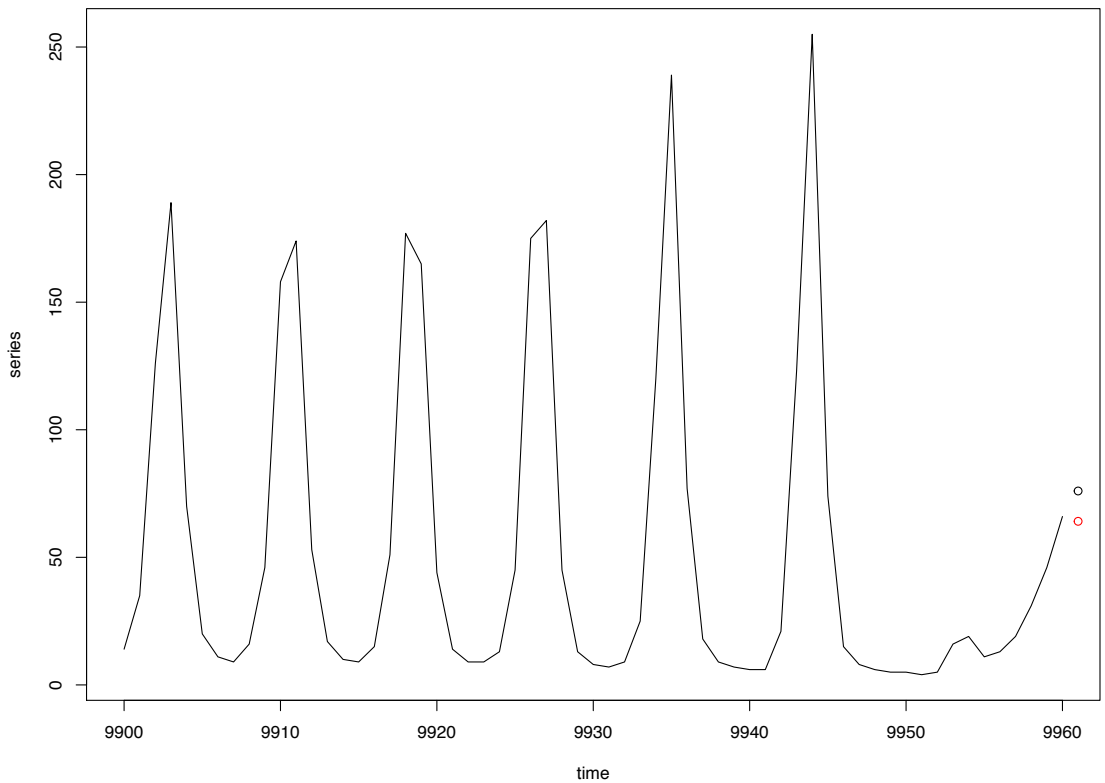
Bias/variance trade-off



Process:	Learner:
Noise $\overset{+}{\rightarrow}$ V	Complexity (e.g. # parameters, VC dim) $\overset{+}{\rightarrow}$ V, $\overset{-}{\rightarrow}$ B
Sample size $\overset{-}{\rightarrow}$ V	Regularisation $\overset{-}{\rightarrow}$ V, $\overset{+}{\rightarrow}$ B
Dimension $\overset{+}{\rightarrow}$ V	Dropout $\overset{-}{\rightarrow}$ V, $\overset{+}{\rightarrow}$ B
Nonlinearity $\overset{+}{\rightarrow}$ B	AutoML (e.g. hyperparameter grid search) $\overset{+}{\rightarrow}$ V, $\overset{-}{\rightarrow}$ B
Nonstationarity $\overset{+}{\rightarrow}$ B	

Forecasting

Univariate one-step-ahead forecasting



Univariate one-step-ahead

- ML **regression plus noise form**

$$\mathbf{y} = f(x) + \mathbf{w}$$

where $\mathbf{w}$ is the noise.

- **Autoregressive formulation**: output is $\mathbf{y} = \mathbf{y}_{t+1}$ and inputs are lagged values

$$y_{t+1} = f\left(\underbrace{y_t, y_{t-1}, \ldots, y_{t-n+1}}_{x}\right) + w_{t+1}$$

- Conventional ML supervised learning machinery may be used to address such task

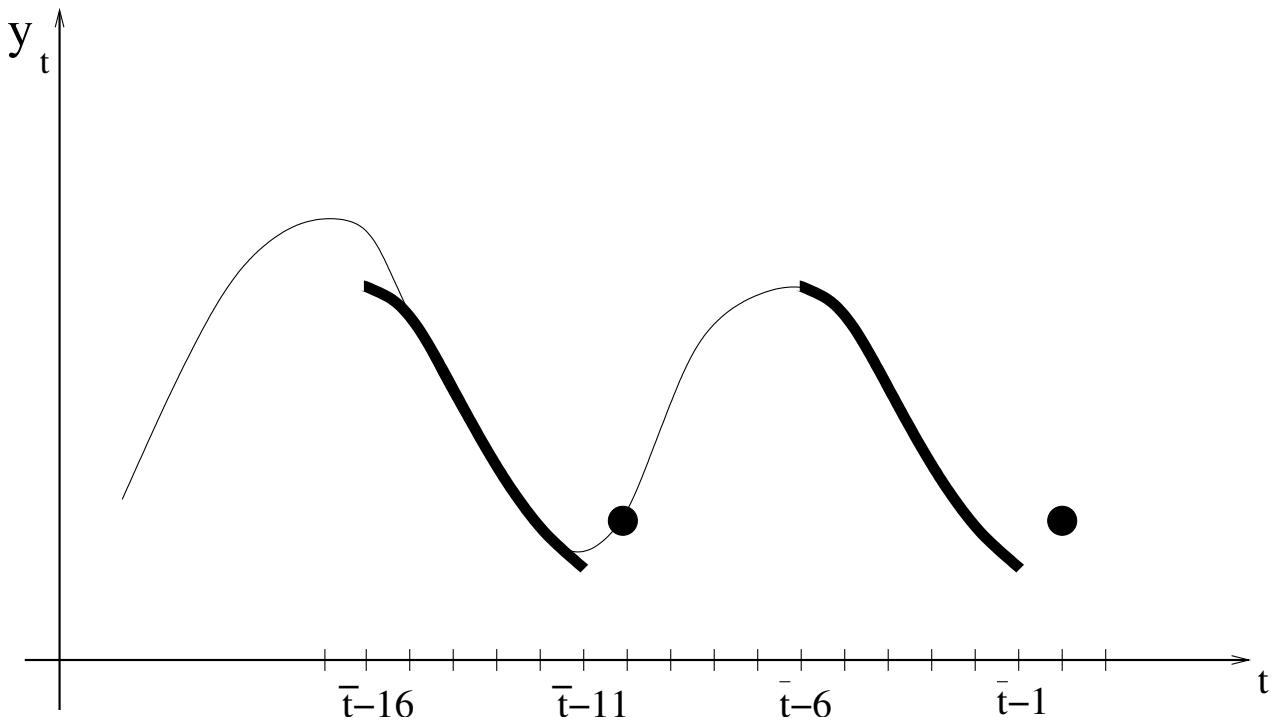
Training data embedding

Given the series $\{y_1, \ldots, y_T\}$ we derive

$$X = \begin{bmatrix} y_{T-1} & y_{T-2} & \cdots & y_{T-n} \\ y_{T-2} & y_{T-3} & \cdots & y_{T-n-1} \\ \vdots & \vdots & \vdots & \vdots \\ y_n & y_{n-1} & \cdots & y_1 \end{bmatrix}, \qquad Y = \begin{bmatrix} y_T \\ y_{T-1} \\ \vdots \\ y_{n+1} \end{bmatrix}$$

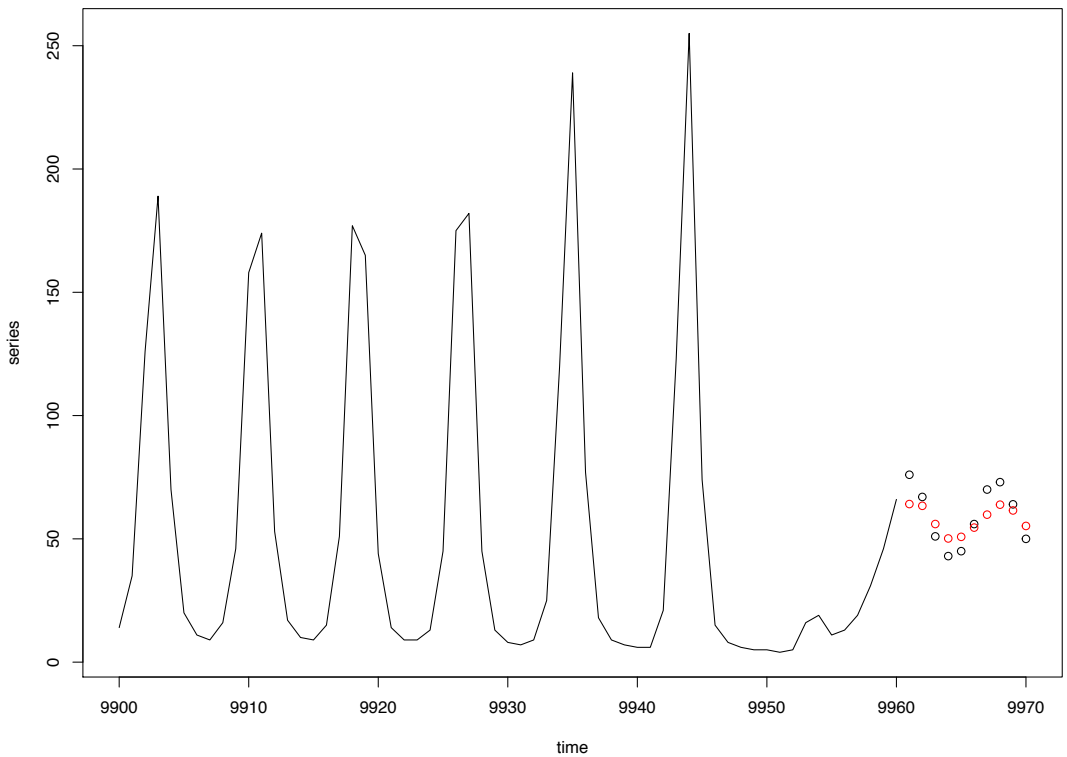
and use them as training set of your preferred learner (FNN, Random Forest, Lazy Learning, Gradient Boosting, ...).

Nearest-neighbor one-step-ahead forecasts



$n = 6$

Multi-step-ahead forecasting



Univariate multi-step-ahead forecasting

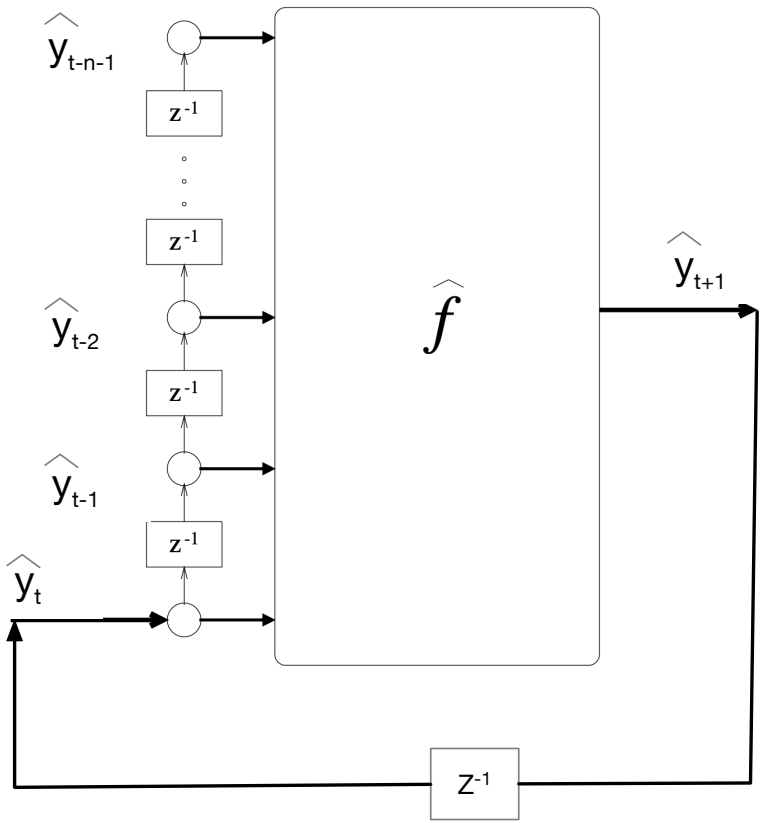
We classify the methods for H-step-ahead prediction according to:

1. *iterated or direct* forecasting
2. *training criterion* (single or multi-step)
3. *single-output* or *multi-output* predictor.

Univariate multi-step-ahead forecasting strategies

1. Iterated: iterates a one-step-ahead predictor with *one-step-ahead training criterion*
2. Iterated: iterates a one-step-ahead predictor with *h_{tr}-step-ahead training criterion* ($1 < h_{tr} \leq H$).
3. Direct: set of independent forecasts at different horizons $t + h, h = 1, \ldots, H$.
4. MIMO: vector of conditionally dependent forecasts

Iterated forecasting



Iterated (or recursive) forecasting: pros/cons

- (+): reuse of conventional supervised ML.
- (+): easy design
- (-): inputs are predicted values instead of actual observations.
- (-) : undesired *accumulation/propagation* of the error.
- (-): low accuracy in long horizon tasks because of one-step-ahead criterion.

Iterated with *h*-step training criterion

- One-step-ahead predictors with **multi-step-ahead cost function**.
- Two approaches
  1. Weight tuning: Recurrent Neural Networks, LSTM
  2. Model selection: Lazy Learning algorithm with bandwidth selection based on multi-step-ahed leave-one-out error (ranked second in the 1998 KULeuven Time Series competition).

Recurrent neural networks

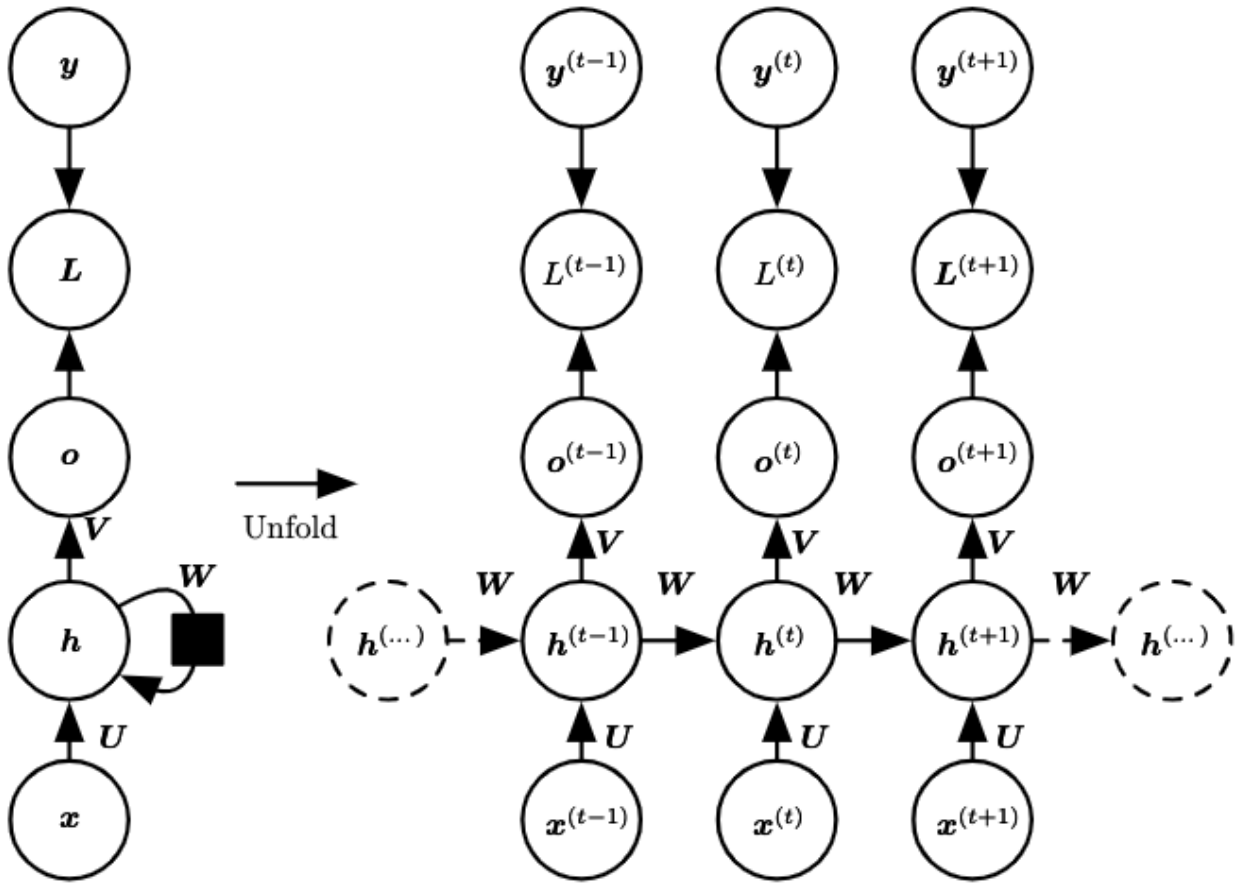
- They date back to (Rumelhart, 1986) and apply neural network learning to sequential data $y_1, \ldots, y_T$
- Recurrent architecture: notion of internal latent state h_t whose dynamics

$$h_t = F(h_{t-1}, x_t, \theta_F) = tanh(b + Wh_{t-1} + Ux_t)$$

underlies the temporal sequence

$$\hat{y}_t = o_t = G(h_t, \theta_G) = c + Vh_t$$

RNN (from Deep Learning book.)



Training data: input/output sequence

$$\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle \ldots \langle x_T, y_T \rangle$$

Backpropagation through time

- BPTT: application of back prop to the unrolled graph
- Very long computational chains due to the recursion, e.g.

$$h_t = Wh_{t-1}$$

- After T steps, if W is not a unit matrix, the term W^T could be ether vanished or exploded.
- For long sequences (beyond 10), BPTT might take a lot to converge.
- Solutions: skip connections, gated units, LSTM

Direct strategy

- It learns **independently** H models f_h

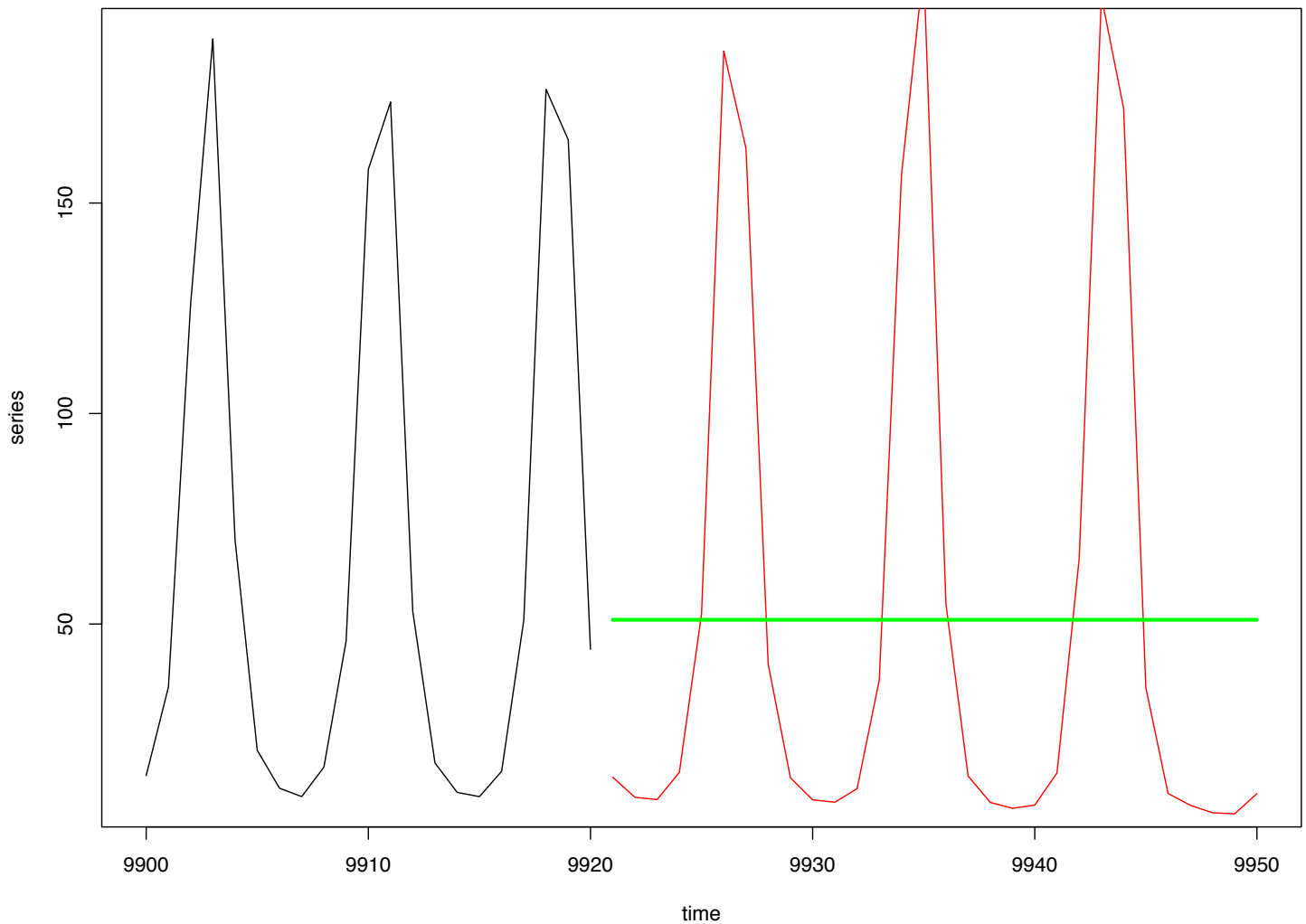
$$y_{t+h} = f_h(y_t, \ldots, y_{t-n+1}) + w_{t+h}$$

  with $t \in \{n, \ldots, N-H\}$ and $h \in \{1, \ldots, H\}$,
- use of conventional supervised learning to estimate H functions f_h,
- multi-step forecast by concatenating the H predictions.

Direct strategy: pros/cons

- (+): reuse of conventional supervised ML.
- (+): no use of approximated values to compute the forecast, then not exposed to error accumulation
- (-) H models are learned independently and no statistical dependencies between the predictions $\hat{y}_{N+h}$ is considered.
- (-) highly nonlinear dependency between values at distant instants.
- (-) large computational time for large horizons.

What is the best continuation?



MIMO (or joint) strategy

- single multiple-output model

$$[y_{t+H}, \ldots, y_{t+1}] = F(y_t, \ldots, y_{t-n+1}) + \mathbf{W}$$

where $t \in \{n, \ldots, N-H\}$, $F : \mathbb{R}^n \to \mathbb{R}^H$ is a vector-valued function, and $\mathbf{W} \in \mathbb{R}^H$ is a noise vector with a covariance that is not necessarily diagonal.

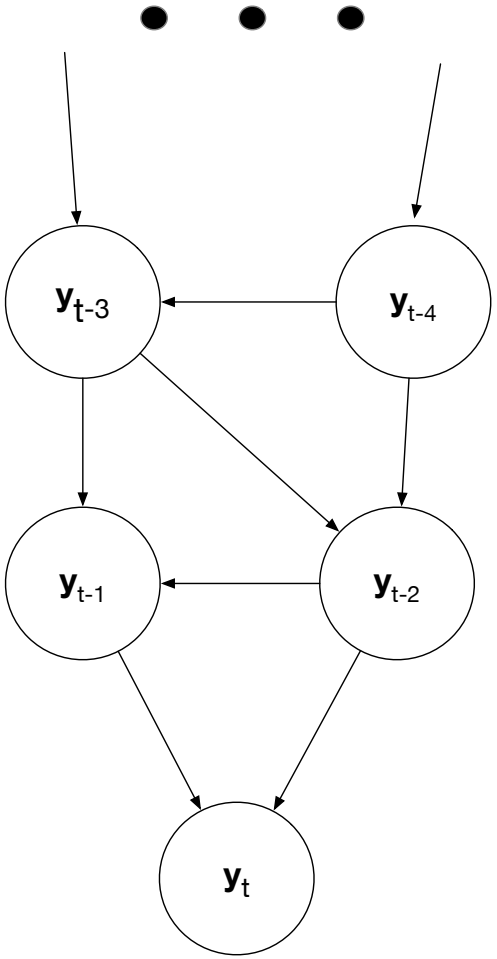
- forecasts returned in a single-step by a multiple-output $\hat{F}$

$$[\hat{y}_{t+H}, \ldots, \hat{y}_{t+1}] = \hat{F}(y_N, \ldots, y_{N-n+1})$$

MIMO strategy

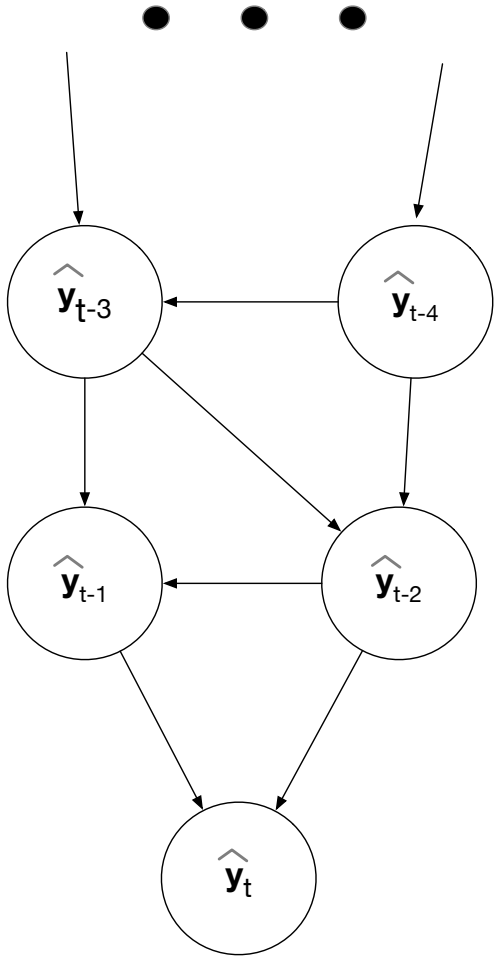
- (+): preserves the stochastic dependency characterizing the time series.
- (+): avoids the conditional independence assumption made by the Direct strategy
- (+): avoids the accumulation of errors which plagues the Recursive strategy.
- (-): multi-task learners
- (-): complex multi-output mapping
- (-) all horizons constrained by the same model structure: variants have been proposed.

Time series dependencies

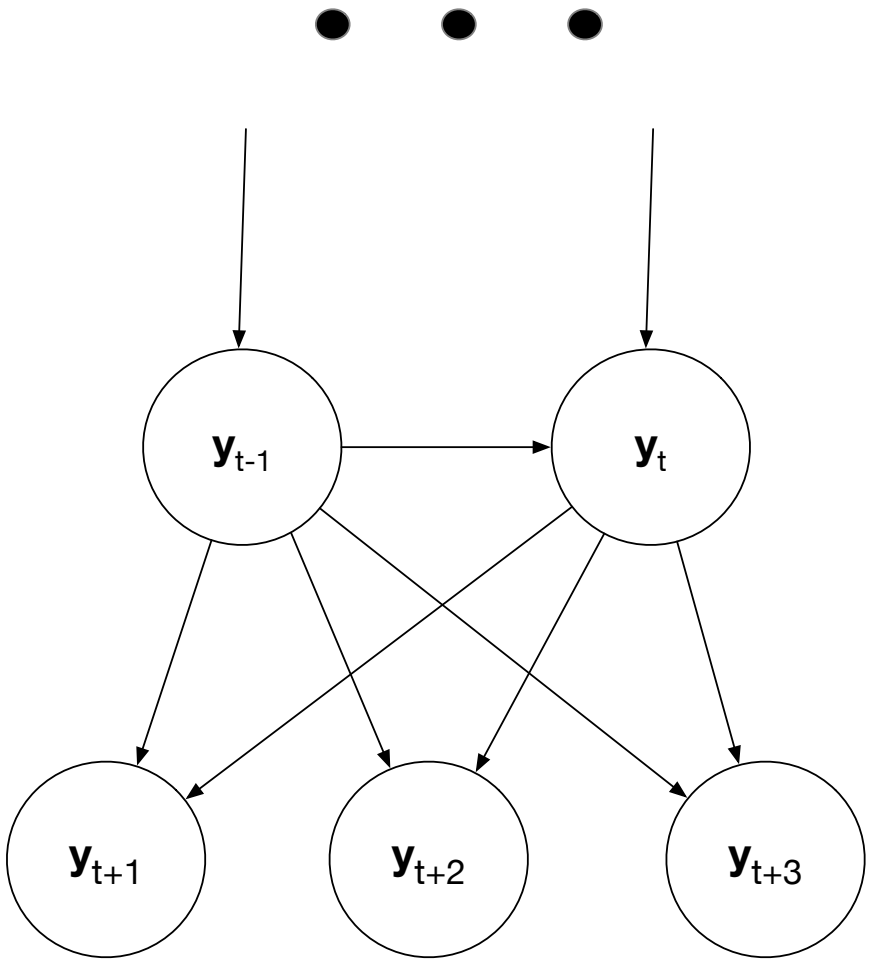


$n = 2$ NAR dependency $y_t = f(y_{t-1}, y_{t-2}) + w(t)$.

Iterated modeling of dependencies

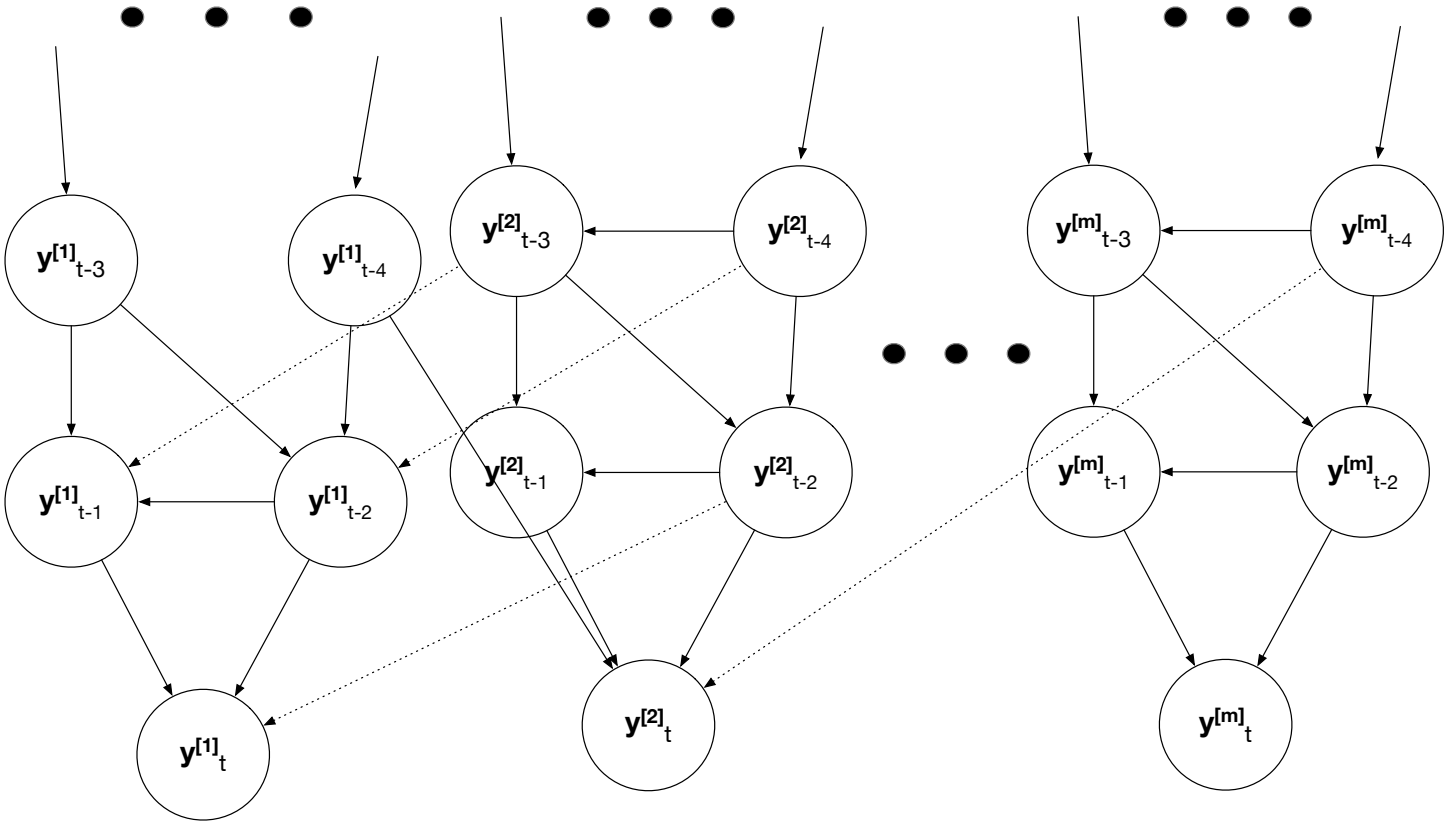


Direct modeling of dependencies



Multivariate forecasting

Multivariate temporal dependency



Temporal and cross-variate dependencies.

Multivariate forecasting strategies

1. Set of m independent univariate tasks.
2. Vector Autoregressive (VAR): linear multivariate version of AR
3. Multi or Single-Output Multi-Inputs forecasting tasks (requiring feature selection or regularization techniques)
4. Recurrent/LSTM Neural Networks
5. Dimension Reduction techniques
    1. Dynamic factor models (DFM)
    2. Autoencoders: nonlinear compression
    3. Partial least squares (PLS): projects both the inputs and the outputs to a new space

Univariate approach

- m independent prediction tasks with horizon H.
- Cross-sectional dependencies are ignored.
- Conventional statistical forecasting (e.g. exponential smoothing, ARIMA) may be used.
- Conventional ML supervised learning may be used as well:

$$y_{t+1}^{[1]} = f_1^{[1]}\left(y_t^{[1]}, \ldots, y_{t-n^{[1]}+1}^{[1]}\right) + w_{t+1}^{[1]}$$

$$\ldots\ldots$$

$$y_{t+H}^{[m]} = f_H^{[m]}\left(y_t^{[m]}, \ldots, y_{t-n^{[m]}+1}^{[m]}\right) + w_{t+H}^{[m]}$$

Multi-input multi-output approach

- Set of multi-input single-output tasks
- Several dependencies to fit: for instance, if $m = 2$ and $H \geq 1$

$$y_{t+1}^{[1]} = f_1^{[1]}\left(y_t^{[1]}, \ldots, y_{t-n^{[1]}+1}^{[1]}, \ldots, y_t^{[m]}, \ldots, y_{t-n^{[m]}+1}^{[m]}\right) + w_{t+1}^{[1]}$$

$$\ldots\ldots$$

$$y_{t+H}^{[m]} = f_H^{[m]}\left(y_t^{[1]}, \ldots, y_{t-n^{[1]}+1}^{[1]}, \ldots, y_t^{[m]}, \ldots, y_{t-n^{[m]}+1}^{[m]}\right) + w_{t+H}^{[m]}$$

- Several hyperparameters: multi-step strategy, functions $f_1^{[1]}, \ldots, f_H^{[m]}$, orders $n^{[1]}, \ldots, n^{[m]}$,...
- Feature extraction, selection and regularization play a major role.

Multi-input data embedding

$$X = \begin{bmatrix} y_{T-1}^{[1]} & \cdots & y_{T-n}^{[1]} & y_{T-1}^{[2]} & \cdots & y_{T-n}^{[2]} & \cdots & y_{T-n}^{[m]} \\ y_{T-2}^{[1]} & \cdots & y_{T-n-1}^{[1]} & y_{T-2}^{[2]} & \cdots & y_{T-n-1}^{[2]} & \cdots & y_{T-n-1}^{[m]} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{n}^{[1]} & \cdots & y_{1}^{[1]} & y_{n}^{[2]} & \cdots & y_{1}^{[2]} & \cdots & y_{1}^{[m]} \end{bmatrix}$$

$$Y = \begin{bmatrix} y_{T}^{[1]} & \cdots y_{T+H-1}^{[1]} & y_{T}^{[2]} & \cdots & y_{T+H-1}^{[m]} \\ y_{T-1}^{[1]} & \cdots y_{T+H-2}^{[1]} & y_{T-1}^{[2]} & \cdots & y_{T+H-2}^{[m]} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{n+1}^{[1]} & \cdots y_{n+H}^{[1]} & y_{n+1}^{[2]} & \cdots & y_{n+H}^{[m]} \end{bmatrix}$$

mn inputs and mH outputs.

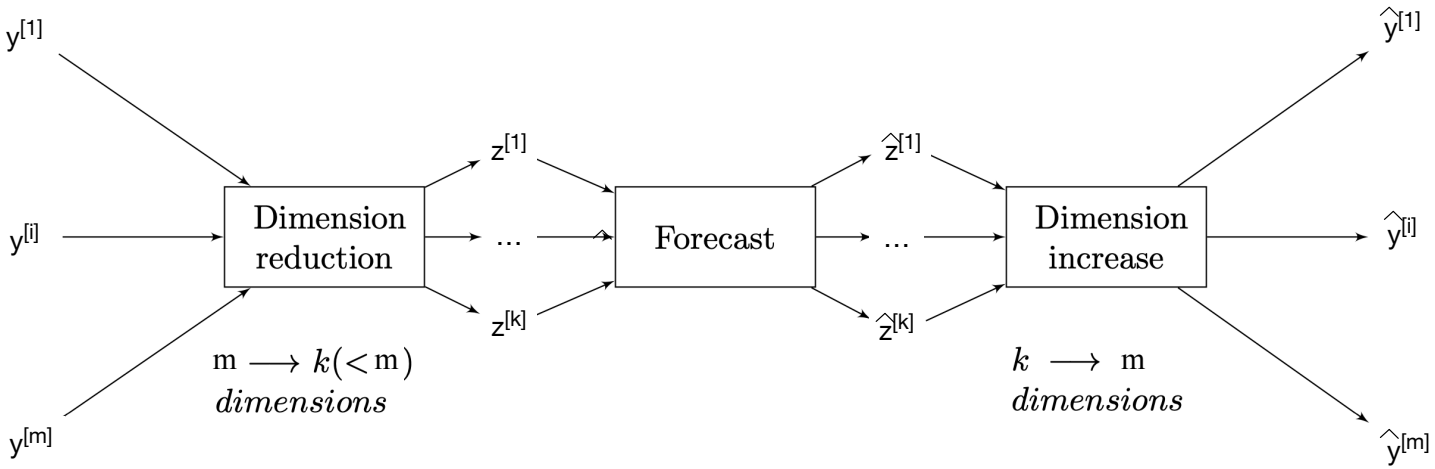
Dynamic factor models

- Basic idea from econometrics: a small number k of unobserved series (or factors) can account for a much larger number n of variables.
- One-step-ahead factor forecasting

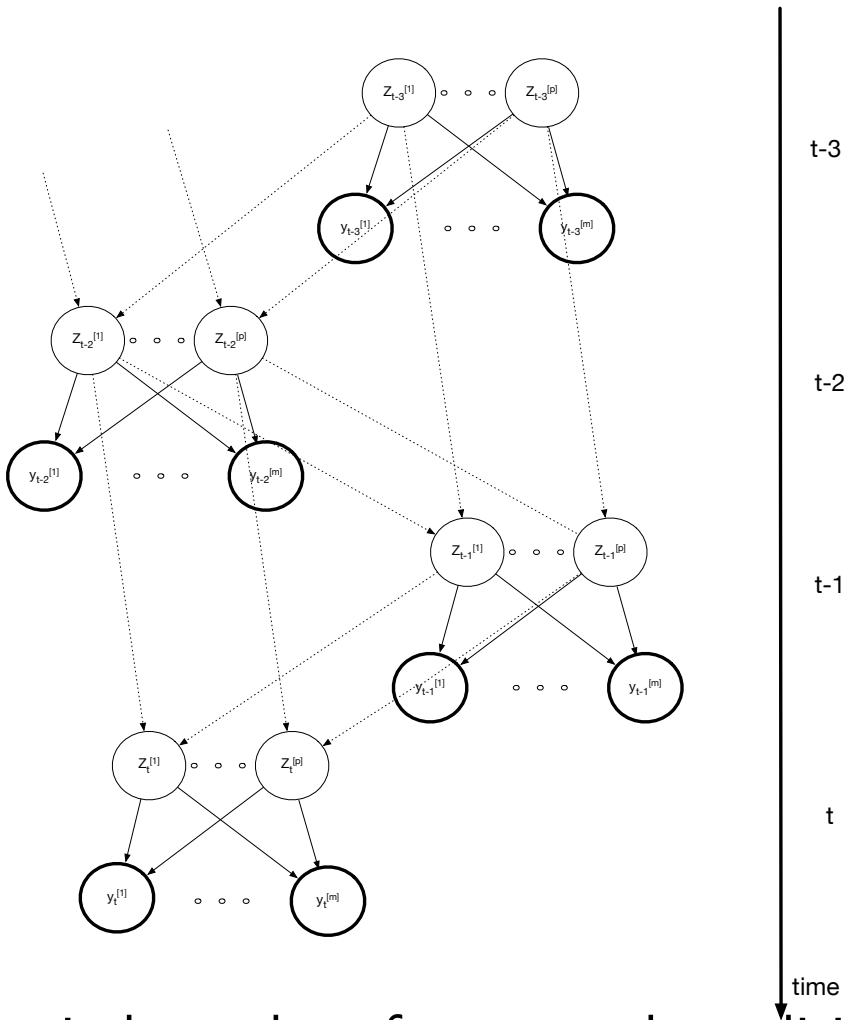
$$Y_{t+1} = WZ_{t+1} + \epsilon_{t+1}$$
$$Z_{t+1} = A_1 Z_t + \cdots + A_n Z_{t-n+1} + \eta_{t+1}$$

  where Z_t is the vector of unobserved factors of size k ($k << m$), A_i are $k \times k$ coefficient matrices, W is the matrix ($m \times k$) of dynamic factor *loadings* and the disturbances terms (also called idiosyncratic disturbances) are assumed to be uncorrelated.
- The latent factors follow a VAR time series process

Dynamic factors



DFM approximation



Two main assumptions: independent factors and conditionally independent observed variables

Multivariate strategies: bias/variance analysis

Possible approaches

- independent univariate tasks
  - (+): low variance, low complexity
  - (-): large bias, neglecting all cross-sectional dependencies
- multi-input multi-output tasks
  - (+): low bias, faithful accounting of dependencies
  - (-): dependent on feature selection (or regularization) strategies
  - (-): large variance, very high complexity (large number of input features and outputs)
- Dynamic factor approach
  - (+): low variance, (small number of factors)
  - (-): large bias, linear model and possible wrong number of latent variables

The DFML forecaster

- Machine learning extension of the DFM:
  1. nonlinear and multi-step-ahead forecaster of the factors
  2. joint selection of the number of the factors and multi-step-ahead strategy.
- Linear (PCA) or nonlinear (autoencoder) technique for dimensionality reduction,
- It forecasts each factor independently using a nonlinear model and a univariate multi-step-ahead forecasting strategy
- Joint selection of the hyperparameters (number of factors, predictor, multi-step-ahead strategy) by using out-of-sample assessment.

Experimental results

Three case studies

- Synthetic cross-sectional and temporal time series: 14 multivariate stochastic processes with cross-sectional and temporal dependencies
- Earth Surface Temperature series: temperature evolution in $n = 200$ countries. Data from Earth Surface Temperature series made available by Berkeley Earth in a Kaggle dataset.
- Volatility series: 7 multivariate volatility proxies derived from $n = 40$ series of the French stock market index CAC40 in the period ranging from 05-01-2009 to 22-10-2014 (almost 6 years)

Cross-sectional and temporal series

$$Y_{t+1}[j] = -0.4\frac{(3 - \bar{Y}_t[\mathcal{N}_j]^2)}{(1 + \bar{Y}_t[\mathcal{N}_j]^2)} + 0.6\frac{3 - (\bar{Y}_{t-1}[\mathcal{N}_j] - 0.5)^3}{1 + (\bar{Y}_{t-1}[\mathcal{N}_j] - 0.5)^4} + W_{t+1}[j]$$

$$Y_{t+1}[j] = (0.4 - 2\exp(-50\bar{Y}_{t-5}[\mathcal{N}_j]^2))\bar{Y}_{t-5}[\mathcal{N}_j] + (0.5 - 0.5\exp(-50\bar{Y}_{t-9}[\mathcal{N}_j]^2))\bar{Y}_{t-9}[\mathcal{N}_j] + W_{t+1}[j]$$

$$Y_{t+1}[j] = (0.4 - 2\cos(40\bar{Y}_{t-5}[\mathcal{N}_j])\exp(-30\bar{Y}_{t-5}[\mathcal{N}_j]^2))\bar{Y}_{t-5}[\mathcal{N}_j] + (0.5 - 0.5\exp(-50\bar{Y}_{t-9}[\mathcal{N}_j]^2))\bar{Y}_{t-9}[\mathcal{N}_j] + W$$

$$Y_{t+1}[j] = 2\exp(-0.1\bar{Y}_t[\mathcal{N}_j]^2)\bar{Y}_t[\mathcal{N}_j] - \exp(-0.1\bar{Y}_{t-1}[\mathcal{N}_j]^2)\bar{Y}_{t-1}[\mathcal{N}_j] + W_{t+1}[j]$$

$$Y_{t+1}[j] = -2\bar{Y}_t[\mathcal{N}_j]I(\bar{Y}_t[\mathcal{N}_j] < 0) + 0.4\bar{Y}_t[\mathcal{N}_j]I(\bar{Y}_t[\mathcal{N}_j] < 0) + W_{t+1}[j]$$

$$Y_{t+1}[j] = 0.8\log(1 + 3\bar{Y}_t[\mathcal{N}_j]^2) - 0.6\log(1 + 3\bar{Y}_{t-2}[\mathcal{N}_j]^2) + W_{t+1}[j]$$

$$Y_{t+1}[j] = 1.5\sin(\pi/2\bar{Y}_{t-1}[\mathcal{N}_j]) - \sin(\pi/2\bar{Y}_{t-2}[\mathcal{N}_j]) + W_{t+1}[j]$$

$$Y_{t+1}[j] = (0.5 - 1.1\exp(-50\bar{Y}_t[\mathcal{N}_j]^2))\bar{Y}_t[\mathcal{N}_j] + (0.3 - 0.5\exp(-50\bar{Y}_{t-2}[\mathcal{N}_j]^2))\bar{Y}_{t-2}[\mathcal{N}_j] + W_{t+1}[j]$$

$$Y_{t+1}[j] = 0.3\bar{Y}_t[\mathcal{N}_j] + 0.6\bar{Y}_{t-1}[\mathcal{N}_j] + \frac{(0.1 - 0.9\bar{Y}_t[\mathcal{N}_j] + 0.8\bar{Y}_{t-1}[\mathcal{N}_j])}{(1 + \exp(-10\bar{Y}_t[\mathcal{N}_j]))} + W_{t+1}[j]$$

N_j: indices of the set of time series which are neighbors of the jth component. $\bar{Y}_t[\mathcal{N}_j]$: average of the value of the neighboring series at time t.

n	H	DFM	DFML$_{PC}$	DFML'$_{PC}$	DFML$_{A}$	DFML'$_{A}$	RNN	DSE	PLS	UNI	VAR	SSA	NAIVE
20	5	0.815	0.813	**0.783**	0.834	0.815	0.793	0.872	0.891	1.012	0.819	0.913	1.913
20	10	0.863	0.851	**0.829**	0.872	0.854	**0.824**	0.925	0.915	1.058	0.62	0.925	1.925
20	20	0.914	0.895	0.875	0.911	0.898	**0.862**	0.957	0.929	1.078	0.909	0.95	1.977
50	5	0.818	0.819	**0.782**	0.842	0.809	0.833	0.890	0.909	1.004	0.821	0.921	1.909
50	10	0.851	0.846	**0.816**	0.868	0.839	0.863	0.906	0.924	1.043	0.850	0.922	1.929
50	20	0.885	0.875	**0.854**	0.895	0.875	0.893	0.923	0.930	1.069	0.881	0.929	1.961
100	5	0.852	0.857	**0.824**	0.909	0.846	0.916	0.922	0.957	1.026	0.913	0.911	1.901
100	10	0.872	0.876	**0.853**	0.924	0.873	0.934	0.958	0.963	1.062	0.908	0.914	1.919
100	20	0.852	0.840	**0.809**	0.883	0.827	0.901	1.028	0.944	1.032	0.861	0.919	1.972
200	5	0.882	0.881	**0.854**	0.942	-	0.956	-	-	1.022	-	-	1.907
200	10	0.895	0.893	**0.872**	0.952	-	0.971	-	-	1.060	-	-	1.928
200	20	0.909	0.908	**0.892**	0.958	-	0.967	-	-	1.086	-	-	1.972
400	5	0.898	0.902	**0.892**	0.984	-	0.985	-	-	-	-	-	1.888
400	10	0.906	0.908	**0.903**	0.991	-	0.994	-	-	-	-	-	1.907
400	20	**0.915**	**0.916**	**0.915**	0.993	-	0.998	-	-	-	-	-	1.949
1000	5	**0.915**	0.919	0.925	1.062	-	-	-	-	-	-	-	1.893
1000	10	**0.919**	0.922	0.930	1.061	-	-	-	-	-	-	-	1.915
1000	20	**0.924**	0.926	0.934	1.058	-	-	-	-	-	-	-	1.958

TABLE II

- NMSE: normalized mean squared error averaged over all the continuation sets
- DFML$_{PC}$: fixed number of factors ($k = 3$) and multi-step-ahead Direct strategy
- DFML'$_{PC}$: automatic selection strategy of the number of factors (in the range $[1, k]$) and the multi- step-ahead strategy (among Direct, Iterated and MIMO).

Earth Surface Temperature

n	H	DFM	DFML$_{PC}$	DFML'$_{PC}$	DFML$_{A}$	DFML'$_{A}$	RNN	PLS	UNI	NAIVE
100	2	**0.099**	**0.111**	**0.099**	0.566	0.594	**0.099**	0.227	0.265	0.692
100	5	0.13	0.151	**0.092**	1.144	0.394	0.102	0.664	0.271	1.981
100	10	0.142	0.164	**0.093**	1.709	0.6	0.113	0.673	0.295	2.247
100	20	0.165	0.173	**0.089**	1.721	0.873	0.11	0.653	0.255	2.165
100	50	0.288	0.187	**0.091**	1.621	0.838	0.111	0.612	0.259	1.894
200	2	**0.124**	0.198	**0.14**	0.7	0.483	0.188	0.49	0.33	0.703
200	5	0.155	0.292	**0.135**	1.131	0.596	0.183	0.834	0.328	1.852
200	10	0.179	0.352	**0.135**.	1.329	0.613	0.202	0.854	0.327	2.125
200	20	0.206	0.381	**0.157**	1.472	0.645	0.229	0.837	0.34	2.038
200	50	0.266	0.405	**0.169**	1.721	0.764	0.242	0.807	0.344	1.801

Volatility

- 7 multivariate volatility proxies from $n = 40$ series of the French stock market index CAC40 from 05-01-2009 to 22-10-2014 (almost 6 years).
- $N = 1489$ OHLC (Opening, High, Low, Closing) and Volume samples for each time series.

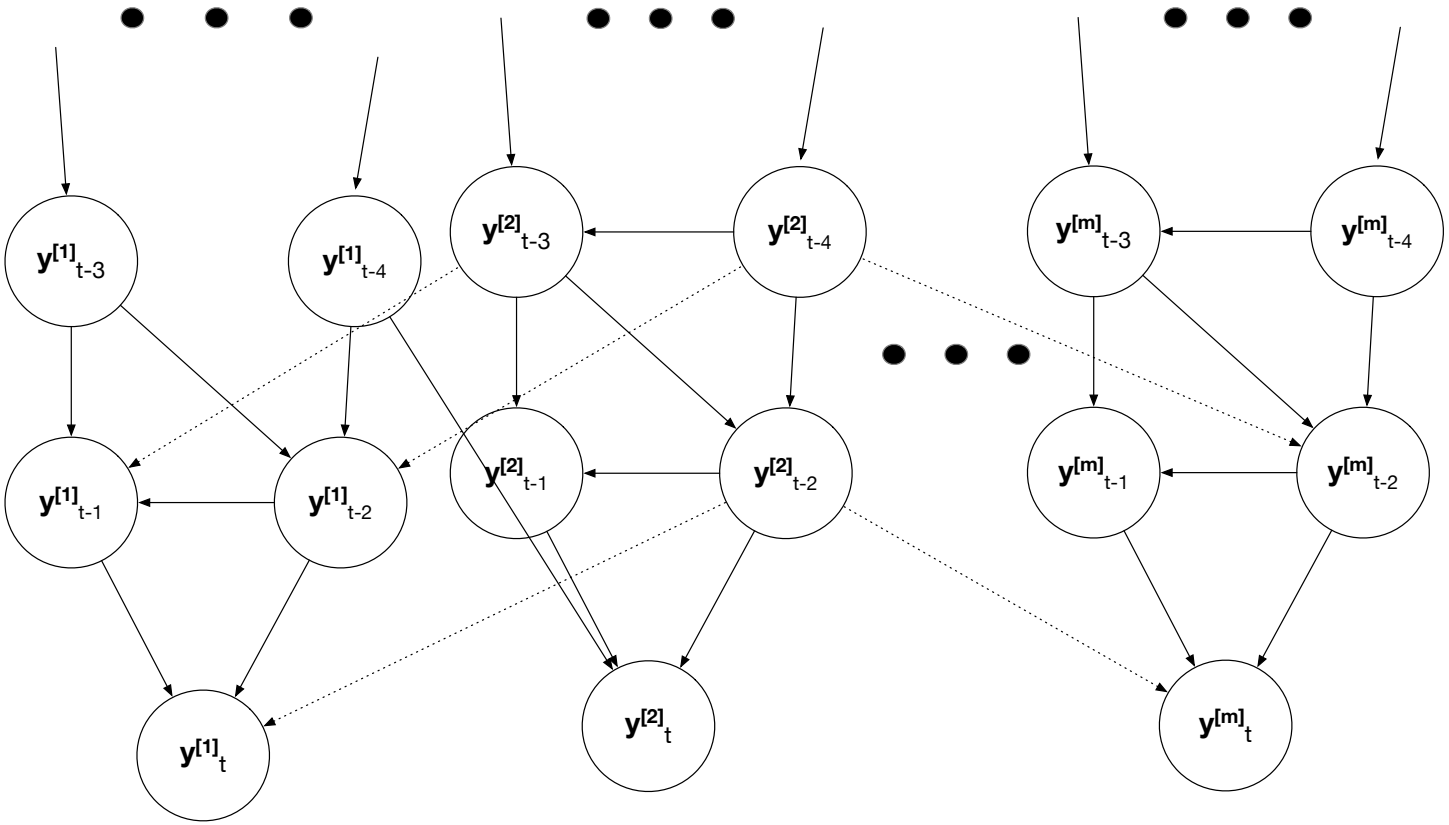
Ind	H	DFM	DFML$_{PC}$	DFML'$_{PC}$	DFML$_{A}$	DFML'$_{A}$	RNN	PLS	UNI	NAIVE
σ_0	2	0.417	0.439	**0.409**	0.462	0.465	0.463	0.416	0.564	0.774
σ_0	5	0.424	0.439	**0.413**	0.463	0.468	0.442	0.421	0.563	0.838
σ_0	10	0.426	0.435	0.454	0.461	0.462	0.439	**0.419**	0.561	0.871
σ_0	20	0.434	0.445	**0.425**	0.465	0.474	0.446	**0.423**	0.573	0.753
σ_0	50	**0.433**	0.449	**0.431**	0.465	0.472	0.443	0.438	0.573	0.759
σ_1	2	**0.362**	0.391	**0.358**	0.422	0.444	0.382	0.369	0.484	0.617
σ_1	5	0.363	0.373	**0.354**	0.416	0.425	0.382	0.364	0.492	0.694
σ_1	10	0.37	0.381	**0.354**	0.414	0.425	0.379	0.361	0.494	0.685
σ_1	20	**0.384**	0.397	**0.383**	0.423	0.433	**0.385**	0.39	0.509	0.718
σ_1	50	0.389	0.411	**0.383**	0.430	0.454	0.390	0.387	0.518	0.647
σ_2	2	**0.305**	0.318	**0.309**	0.384	0.406	0.333	0.321	0.41	0.518
σ_2	5	0.31	0.317	**0.304**	0.380	0.394	0.349	0.316	0.404	0.553
σ_2	10	0.324	0.323	**0.3**	0.376	0.389	0.347	0.316	0.407	0.522
σ_2	20	0.35	**0.343**	**0.354**	0.385	0.416	0.360	**0.338**	0.438	0.534
σ_2	50	0.375	0.383	**0.328**	0.402	0.421	0.399	**0.326**	0.493	0.543

Volatility

Ind	H	DFM	DFML$_{PC}$	DFML'$_{PC}$	DFML$_{A}$	DFML'$_{A}$	RNN	PLS	UNI	NAIVE
σ_3	2	**0.322**	0.335	0.332	0.426	0.441	0.356	0.341	0.407	0.507
σ_3	5	**0.325**	0.334	**0.323**	0.419	0.433	0.363	0.336	0.416	0.587
σ_3	10	0.338	0.345	**0.328**	0.420	0.431	0.364	0.337	0.422	0.587
σ_3	20	0.364	**0.367**	**0.354**	0.433	0.445	0.379	0.36	0.453	0.59
σ_3	50	0.386	0.388	**0.344**	0.436	0.460	0.403	**0.345**	0.506	0.561
σ_4	2	**0.3**	0.312	0.314	0.389	0.412	0.332	0.318	0.392	0.523
σ_4	5	**0.304**	**0.319**	**0.309**	0.388	0.397	0.326	0.316	0.396	0.567
σ_4	10	0.319	0.331	**0.302**	0.385	0.406	0.340	0.315	0.4	0.511
σ_4	20	0.344	**0.34**	**0.328**	0.388	0.426	0.359	**0.329**	0.427	0.494
σ_4	50	0.373	0.386	0.329	0.405	0.402	0.388	**0.322**	0.504	0.536
σ_5	2	**0.299**	0.311	0.312	0.389	0.412	0.329	0.317	0.391	0.521
σ_5	5	**0.304**	**0.317**	**0.304**	0.387	0.400	0.341	0.316	0.394	0.564
σ_5	10	0.319	0.327	**0.301**	0.387	0.402	0.336	0.315	0.398	0.51
σ_5	20	0.345	0.341	**0.309**	0.389	0.430	0.352	0.329	0.426	0.495
σ_5	50	0.373	0.383	0.328	0.395	0.410	0.405	**0.322**	0.504	0.535
σ_6	2	**0.297**	0.312	0.312	0.385	0.428	0.322	0.325	0.385	0.506
σ_6	5	**0.299**	0.309	**0.302**	0.380	0.415	0.334	0.314	0.39	0.554
σ_6	10	0.312	0.32	**0.296**	0.381	0.423	0.356	0.313	0.4	0.507
σ_6	20	0.336	0.34	**0.319**	0.386	0.428	0.351	0.327	0.424	0.491
σ_6	50	0.365	0.382	0.324	0.401	0.433	0.401	**0.319**	0.494	0.528

Causal inference

Association vs causality



$y^{[1]}$ and $y^{[m]}$ are dependent but none of them is a cause of the other.

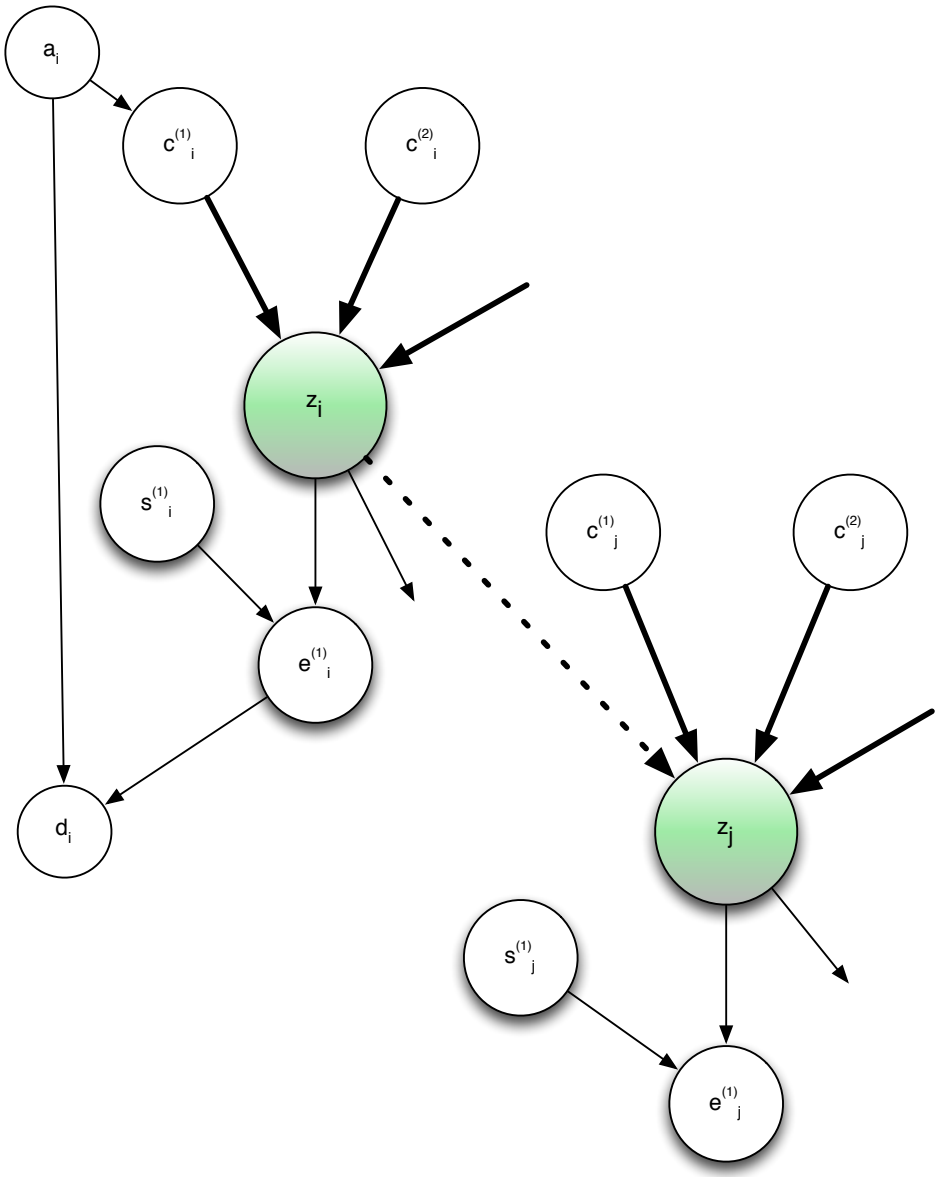
Causal inference and time series

- Aim: discriminate between associative dependencies and effective causal relationships in observational data.
- Highly challenging in large-variate and temporal settings (e.g. in spatio-temporal time series) where the multivariate nature of interactions induces a significant correlation between most of the variables.
- Conventional algorithms relies on conditional independence tests or maximum likelihood optimisation.
- D2C approach rationales:
  - dependency is symmetric, causality is not
  - causality leaves footprints in distributions
  - a machine learning strategy may be used to reduce causal indistinguishability.

The D2C approach

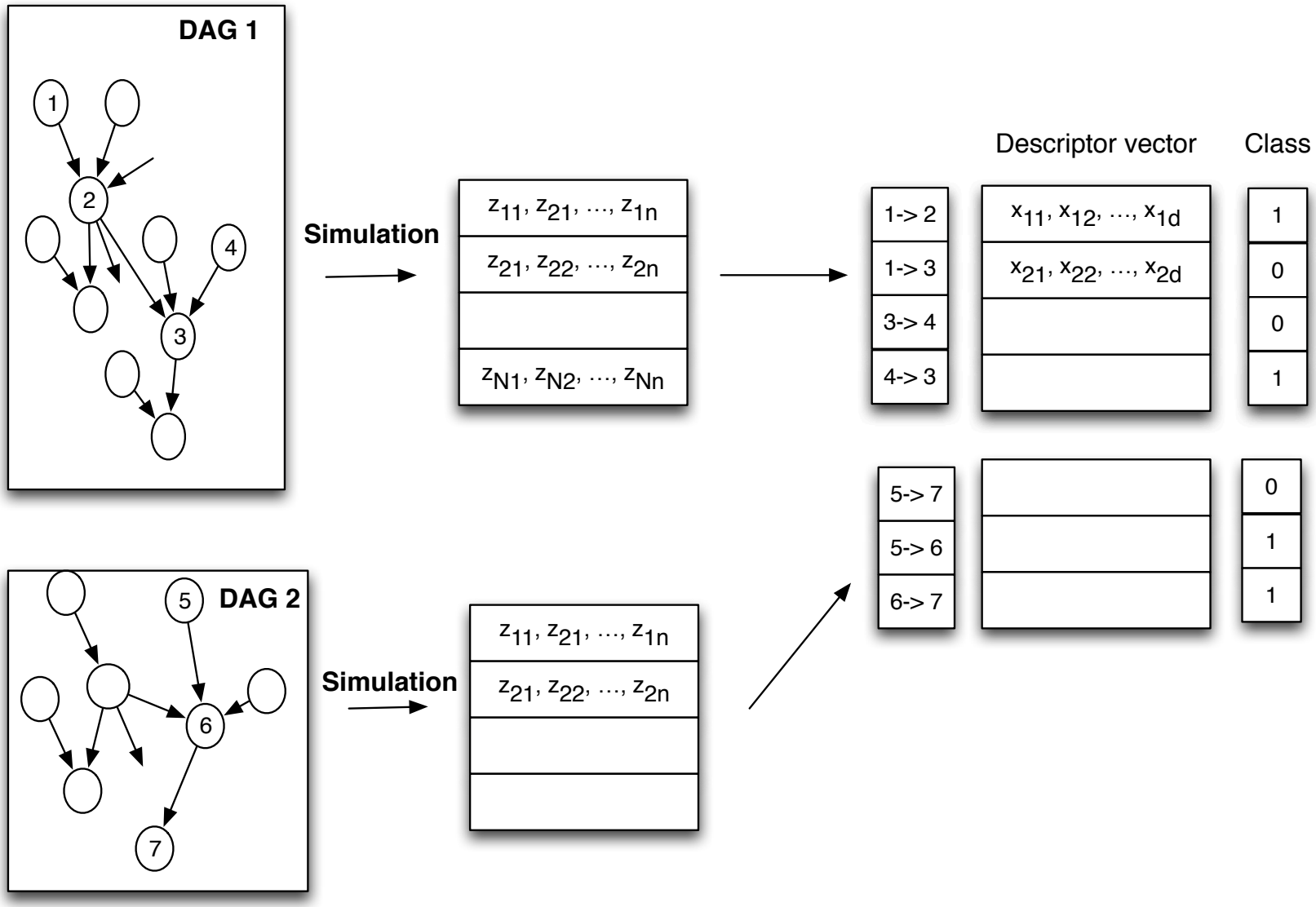
- Given two variables, the D2C approach infers from a number of observed statistical features of the n-variate distribution the probability of the existence of a directed causal link.
- **Causal inference as a supervised learning task** where inputs are features describing the probabilistic dependency and the output is a class denoting the existence of the causal link.
- Once sufficient training data are made available, conventional feature selection algorithms and classifiers can be used to return a prediction.

Causality and asymmetric descriptors

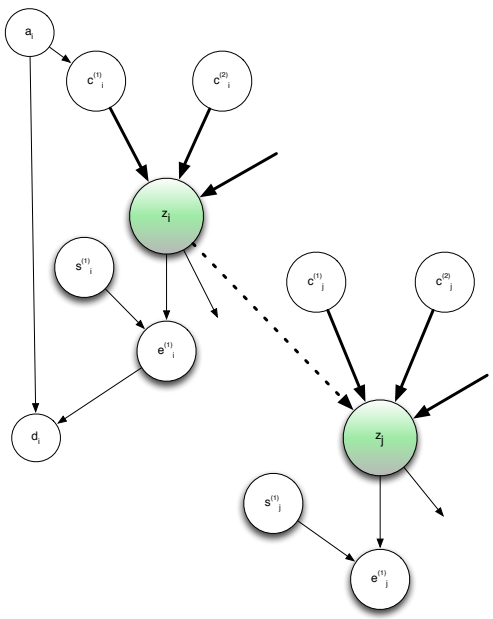


Relation i, j	Relation j, i
$\forall k \quad \mathbf{z}_i \not\perp\!\!\!\perp \mathbf{c}_j^{(k)} \vert \mathbf{z}_j$	$\forall k \quad \mathbf{z}_j \perp\!\!\!\perp \mathbf{c}_i^{(k)} \vert \mathbf{z}_i$
$\forall k \quad \mathbf{e}_i^{(k)} \not\perp\!\!\!\perp \mathbf{c}_j^{(k)} \vert \mathbf{z}_j$	$\forall k \quad \mathbf{e}_j^{(k)} \perp\!\!\!\perp \mathbf{c}_i^{(k)} \vert \mathbf{z}_i$
$\forall k \quad \mathbf{c}_i^{(k)} \not\perp\!\!\!\perp \mathbf{c}_j^{(k)} \vert \mathbf{z}_j$	$\forall k \quad \mathbf{c}_j^{(k)} \perp\!\!\!\perp \mathbf{c}_i^{(k)} \vert \mathbf{z}_i$
$\forall k \quad \mathbf{z}_i \perp\!\!\!\perp \mathbf{c}_j^{(k)}$	$\forall k \quad \mathbf{z}_j \not\perp\!\!\!\perp \mathbf{c}_i^{(k)}$

D2C training phase



The D2C approach



- Training phase:
  1. generate and simulate a large number of Bayesian networks.
  2. for a number of edges, measure a number of asymmetric descriptors and the corresponding label (e.g. node 1 parent of node 2).
  3. train a classifier (e.g. a Random Forest) returning the probability of a causal link given the descriptors value.
- Prediction phase: given a dataset and two variables of interest
  1. estimate the Markov Blankets of the two variables of interest and ranks its components in terms of their causal nature,
  2. compute a number of asymmetric descriptors and
  3. return the classifier prediction.

Context-aware D2C

1. it ranks the most relevant variables for $\mathbf{z}_i$ and $\mathbf{z}_j$ into the sets $\mathbf{M}_i$ and $\mathbf{M}_j$.
2. for each pairs $(\mathbf{m}_i^{(k)}, \mathbf{m}_j^{(k)})$, where $\mathbf{m}_i^{(k)} \in \mathbf{M}_i$ and $\mathbf{m}_j^{(k)} \in \mathbf{M}_j$, it computes (conditional) mutual information descriptors

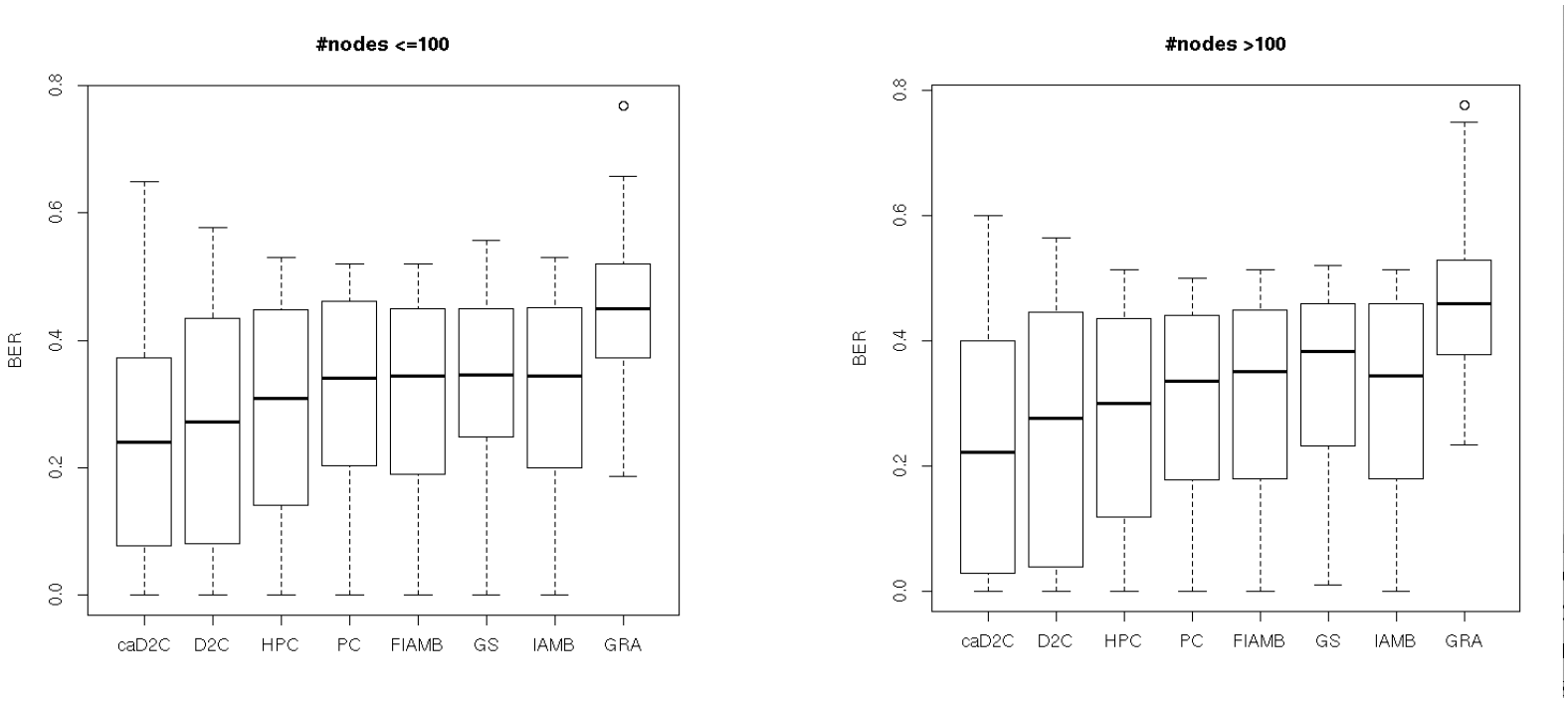
$$d_1^{(k)}(i,j) = I(\mathbf{z}_i; \mathbf{m}_j^{(k)} | \mathbf{z}_j), \quad d_2^{(k)}(i,j) = I(\mathbf{m}_i^{(k)}; \mathbf{m}_j^{(k)} | \mathbf{z}_j),$$
$$d_3^{(k)}(i,j) = I(\mathbf{m}_i^{(k)}; \mathbf{m}_j^{(k)} | \mathbf{z}_j), d_4^{(k)}(i,j) = I(\mathbf{z}_j; \mathbf{m}_i^{(k)})$$

3. for each pairs $(\mathbf{m}_i^{(k)}, \mathbf{m}_i^{(t)})$ and $(\mathbf{m}_i^{(k)}, \mathbf{m}_j^{(t)})$, where $\mathbf{m}_i^{(k)}, \mathbf{m}_i^{(t)} \in \mathbf{M}_i$ and $\mathbf{m}_j^{(k)}, \mathbf{m}_j^{(t)} \in \mathbf{M}_j$, it computes the context aware interaction information descriptor
4. it computes a set of quantiles of the empirical distributions of the terms computed in the two steps before and use them as input vector of the D2C classifier.

Experiments

- Causal inference experiments on a large number of simulated stationary time series characterized by nonlinearity, large dimension and cross-sectional dependencies.
- Benchmarking against D2C and state-of-the-art causal inference algorithms:
  - Semi-Interleaved HITON-PC local discovery structure learning algorithms (HPC)
  - incremental association MB constraint-based structure learning algorithm (IAMB)
  - Fast-IAMB version of IAMB (FIAMB)
  - Grow-Shrink (GS)
  - PC from `pcalg` package (PCalg)
  - Granger test (GRA) from `lmtest` package

Results BER (the lower the better!)



Distribution of the BER accuracy for the 500 test time series. Above (below) we report the BER distribution over the time series whose associated DAG has less (more) than 100 nodes.

Forecasting: conclusions

- "We are drowning in data and starving for knowledge"
- In the era of big data the Internet of Things (IoTs) technology produces massive amounts of heterogeneous multidimensional temporal data in real time.
- Multivariate and multistep prediction is the most difficult prediction problem conceivable in forecasting.
- Need to address jointly multiple bias/variance issues originating from different aspects of the task (nonlinearity, dimensionality, large horizon, multiple dependencies, noise)
- No easy one-fits-all solution!

Causal inference: conclusions

- "We are drowning in associations and starving for causality"
- Sometimes accurate prediction is not sufficient: we want to understand the causal mechanism.
- Big data expose our society to a number of (real or presumed) associations that could have impact on lifestyle, health choices, economic and political decisions.
- Pessimistic point of view: *Correlation (or dependency) does not imply causation.*
- Optimistic point of view: *Causation implies correlation (or dependency).*
- **Causality leaves footprints on the patterns of stochastic dependency which can be (hopefully) retrieved from data**.

Acknowledgments

Joint work with:

- G. Paldino (ULB MLG),
- F. De Caro (USannio, I)
- S. Ben Taieb (UMONS, B),
- J. De Stefani (Delft, NL).