High-dimensional modeling and forecasting for wind power generation

Jakob Messner*, Pierre Pinson*, Yongning Zhao†,*

*Technical University of Denmark, †China Agricultural University
(authors in alphabetical order)

Contact - email: ppin@elektro.dtu.dk - webpage: www.pierrepinson.com

YEQT Winter School on Energy Systems - 13 December 2017

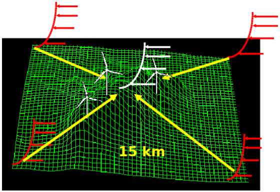
Outline

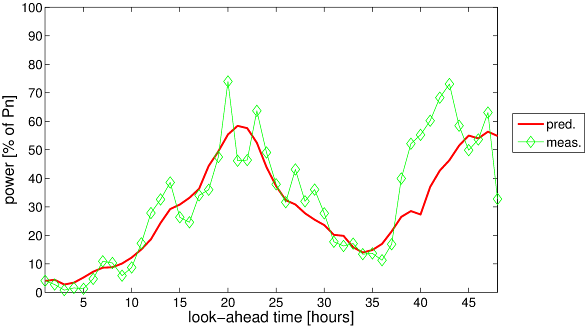
- **Motivations for high-dimension learning and forecasting**
- **General sparsity control for VAR models**
- **Online sparse and adaptive learning for VAR models**
- **Distributed learning**
- **Outlook**

1. **From single wind farms to entire regions (1000s)**

A traditional view on wind power forecasting

The wind power forecasting problem is defined for a single location...





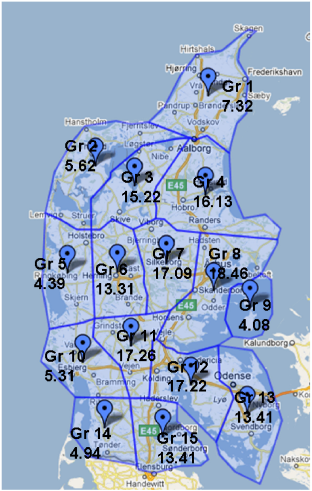
... or, if several locations, by considering each of them individually

(Note that, for simplicity, we will only look at very short-term forecasting in this talk, i.e., from a few mins to 1-hour ahead)

Wind farms as a network of sensors

Many works showed that **forecast quality could be significantly improved:**

- by using data at offsite locations (i.e., other wind farms)
- based on spatio-temporal modelling (and the likes)

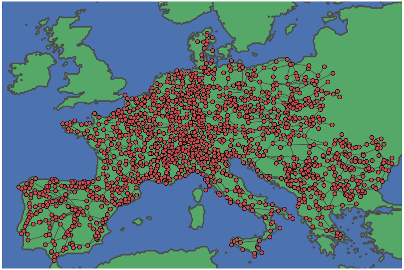


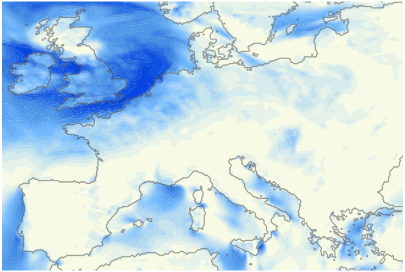
improvement of 1-hour ahead forecast RMSE

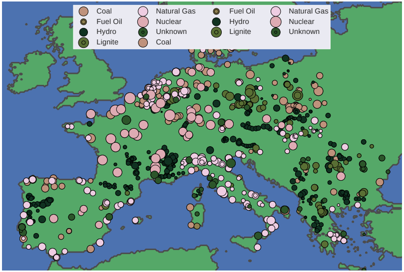
- A Danish example...
- Accounting for spatio-temporal effects allows for the correction of aggregated power forecasts **for horizons up to 8 hours ahead**
- Largest improvements at **horizons of 2-5 hours ahead**

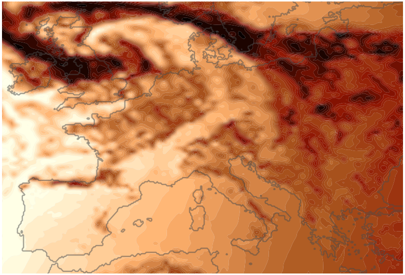
Scaling it up

Ultimately, we would like to predict all wind power generation, also solar and load, at the scale of a continental power system, e.g. the European one





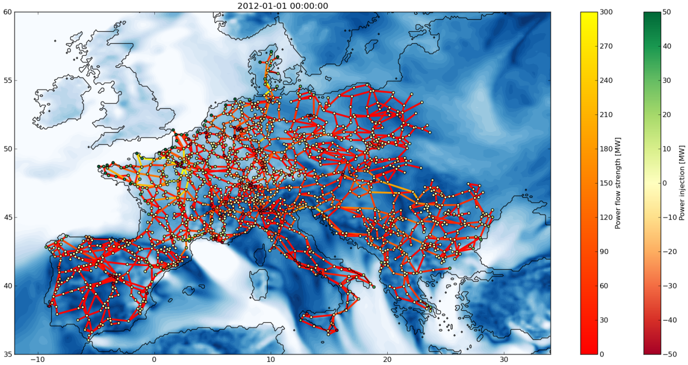




RE-Europe dataset, available at `zenodo.org`, descriptor in *Nature, Scientific Data*

The big picture...

- The **"grand forecasting challenge"**: predict *renewable power generation*, *dynamic uncertainties* and *space-time dependencies* at once for the whole Europe...!



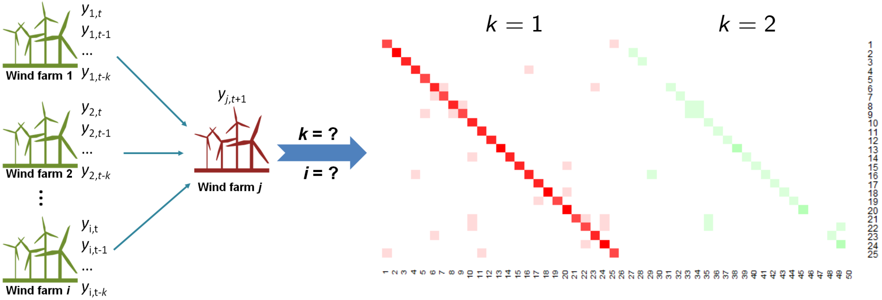
- **Linkage with future electricity markets:**
  - Monitoring and forecasting of the complete **"Energy Weather"** over Europe
  - Provides all necessary information for coupling of various existing markets (e.g., day-ahead, balancing), and deciding upon optimal cross-border exchanges

2. **A proposal for general sparsity control (not online though)**

Sparsity-controlled vector autoregressive (SC-VAR) model

Traditional LASSO-VAR can **only provide overall sparse solutions**, but **not allow for fine-tuning different aspects of sparsity**, e.g. :

- overall number of nonzero coefficients of VAR (S_A), *i.e. the LASSO-VAR*
- number of explanatory wind farms used in VAR to explain target wind farm i (S_F^i)
- number of past observations of each explanatory wind farm to explain target wind farm i (S_P^i)
- number of nonzero coefficients to explain target wind farm i (S_N^i).



These aspects can be used to control the sparse structure of the solution as needed, especially **when prior knowledge on spatio-temporal characteristics of wind farms are available for sparsity-control and expected to improve the forecasting**.

How to freely control the sparse structure... [E. Carrizosa, et al. 2017]

- Introducing binary control variables γ_j^i and δ_{jk}^i
  - γ_j^i controls whether wind farm j is used to explain target wind farm i.
  - δ_{jk}^i controls whether the coefficient α_{jk}^i is zero or not.
- Reformulating the VAR estimation as a constrained mixed integer non-linear programming (MINLP) problem.

For example: $N = 3$ wind farms, VAR(2) with $p = 2$ lags

$$\begin{bmatrix} \gamma_1^1 & \gamma_2^1 & \gamma_3^1 \\ \gamma_1^2 & \gamma_2^2 & \gamma_3^2 \\ \gamma_1^3 & \gamma_2^3 & \gamma_3^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \Longleftrightarrow \mathbf{A} = \begin{bmatrix} \alpha_{11}^1 & 0 & \alpha_{31}^1 & \alpha_{12}^1 & 0 & \alpha_{32}^1 \\ 0 & \alpha_{21}^2 & 0 & 0 & \alpha_{22}^2 & 0 \\ \alpha_{11}^3 & 0 & \alpha_{31}^3 & \alpha_{12}^3 & 0 & \alpha_{32}^3 \end{bmatrix}$$

If additionally with control variable $\delta_{11}^3 = 0$, then

$$\mathbf{A} = \begin{bmatrix} \alpha_{11}^1 & 0 & \alpha_{31}^1 & \alpha_{12}^1 & 0 & \alpha_{32}^1 \\ 0 & \alpha_{21}^2 & 0 & 0 & \alpha_{22}^2 & 0 \\ 0 & 0 & \alpha_{31}^3 & \alpha_{12}^3 & 0 & \alpha_{32}^3 \end{bmatrix}$$

That is:

$$\gamma_j^i = 0 \Leftrightarrow \sum_{k=1}^{p} \delta_{jk}^i = 0 \qquad \delta_{jk}^i = 0 \Leftrightarrow \alpha_{jk}^i = 0$$

Sparsity-controlled vector autoregressive (SC-VAR) model

$$\min_{\alpha,\delta,\gamma} \quad \sum_{i=1}^{N}\sum_{t=p}^{T}\left(y_{i,t+1}-\sum_{j=1}^{N}\sum_{k=1}^{p}\alpha_{jk}^{i}y_{j,t-k+1}\right)^{2}$$

subject to

$$\delta_{jk}^{i} \le \gamma_{j}^{i}, \forall k \in \mathbf{K}, i, j \in \mathbf{I}$$

$$\sum_{j=1}^{N}\gamma_{j}^{i} \le S_{F}^{i}, \forall i \in \mathbf{I}$$

$$\sum_{k=1}^{p}\gamma_{j}^{i}\delta_{jk}^{i} \le S_{P}^{i}, \forall i, j \in \mathbf{I}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{p}\delta_{jk}^{i} \le S_{A}, \forall k \in \mathbf{K}, i, j \in \mathbf{I}$$

$$\sum_{j=1}^{N}\sum_{k=1}^{p}\delta_{jk}^{i} \le S_{N}^{i}, \forall i \in \mathbf{I}$$

$$\left|\alpha_{jk}^{i}\right| \ge \eta_{j}^{i}\delta_{jk}^{i}, \forall k \in \mathbf{K}, i, j \in \mathbf{I}$$

$$\alpha_{jk}^{i}(1-\delta_{jk}^{i}) = 0, \forall k \in \mathbf{K}, i, j \in \mathbf{I}$$

$$\delta_{jk}^{i}, \gamma_{j}^{i} \in \{0,1\}, \forall k \in \mathbf{K}, i, j \in \mathbf{I}$$

- $\mathbf{I} = \{1, 2, \cdots, N\}$
- $\mathbf{K} = \{1, 2, \cdots, p\}$
- S_A- overall number of nonzero coefficients of VAR
- S_F^i- number of explanatory wind farms used in VAR to explain target wind farm i
- S_P^i- number of past observations of each explanatory wind farm to explain target wind farm i
- S_N^i- number of nonzero coefficients to explain target wind farm i
- η_j^i- a threshold requires that only coefficients with absolute value greater than or equal to η_j^i are effective otherwise will be zero.

Pros and cons of SC-VAR model

Pros

- allows for fully controlling the sparsity from different aspects.
- can be directly solved by off-the-shelf standard MINLP solvers.

Cons

- *SC-VAR allows for sparsity-control but doesn't tell how to control*. No information is available for setting so many parameters, which are practically intractable when dealing with high dimensional wind power forecasting.
- The constraint $\sum_{k=1}^{p} \gamma_j^i \delta_{jk}^i \leq S_P^i$ is nonlinear.
- The constraints are redundant: $S_F^i + S_P^i = S_N^i$, $\sum_{i \in \mathbf{I}} S_N^i = S_A$
- The constraint $\sum \sum \sum \delta_{jk}^i \leq S_A$ makes the optimization problem non-decomposable, which slows down the computation.
- Too many variables to be optimized: VAR coefficients α_{jk}^i, binary control variables γ_j^i and δ_{jk}^i.

(Note that, though $\left|\alpha_{jk}^i\right| \geq \eta_j^i \delta_{jk}^i$ *and* $\alpha_{jk}^i(1-\delta_{jk}^i) = 0$ *are also nonlinear, [E. Carrizosa, et al. 2017] provides linearized reformulation for them.)*

Correlation-constrained SC-VAR (CCSC-VAR) model

Incorporate explicit spatial correlation information into the constraints!

$$\min_{\alpha,\delta} \quad \sum_{i=1}^{N}\sum_{t=p}^{T}\left(y_{i,t+1}-\sum_{j=1}^{N}\sum_{k=1}^{p}\alpha_{jk}^{i}y_{j,t-k+1}\right)^2$$

$$\text{subject to} \quad \delta_{jk}^{i} \le \lambda_j^i, \forall k \in \mathbf{K}, i,j \in \mathbf{I}$$

$$\sum_{k=1}^{p}\delta_{jk}^{i} \ge \lambda_j^i, \forall i,j \in \mathbf{I}$$

$$\sum_{j=1}^{N}\sum_{k=1}^{p}\delta_{jk}^{i} \le S_N^i, \forall i \in \mathbf{I}$$

$$\left|\alpha_{jk}^{i}\right| \le M \cdot \delta_{jk}^{i}, \forall k \in \mathbf{K}, i,j \in \mathbf{I}$$

$$\delta_{jk}^{i}, \gamma_j^i \in \{0,1\}, \forall k \in \mathbf{K}, i,j \in \mathbf{I}$$

where

$$\lambda_j^i = \begin{cases} 1, \phi_j^i \ge \tau \\ 0, \phi_j^i < \tau \end{cases}$$

$$\left|\alpha_{jk}^{i}\right| \le M \cdot \delta_{jk}^{i} \Leftrightarrow \begin{cases} -M \le \alpha_{jk}^{i} \le M, \delta_{jk}^{i} = 1 \\ \alpha_{jk}^{i} = 0, \delta_{jk}^{i} = 0 \end{cases}$$

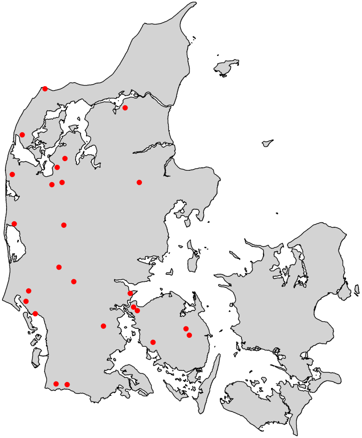
Notations:

- ϕ_j^i is the Pearson correlation between wind farms i and j.
- M is a positive constant number (Generally $M < 2$).
- τ and S_N^i are used to control sparsity.

Improvements: *(simpler but better!)*

- Less parameters need to be tuned while the sparsity-control ability is preserved.
- More capable of characterizing the true inter-dependencies between wind farms.
- Less variables to be optimized.
- All constraints are linear.
- The model is decomposable.

Application and case study



- 25 wind farms randomly chosen over western Denmark
- 15-minute resolution
- 20.000 data points for each wind farm

Compared Models:

- **Local forecasting models**
  - Persistence method
  - Auto-Regressive model
- **Spatio-temporal models**
  - VAR model
  - LASSO-VAR model
  - SC-VAR model
  - CCSC-VAR model

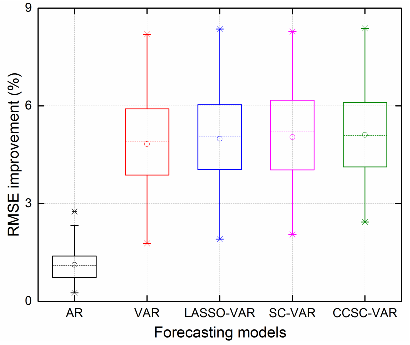
Performance Metrics:

- Root Mean Square Error (RMSE)
- Mean Absolute Error (MAE)
- Sparsity for spatial models

Application and case study

Table: The average RMSE and MAE for all 25 wind farms for different forecasting models

Metrics	Persistence	AR	VAR	LASSO-VAR	SC-VAR	CCSC-VAR
Average RMSE	0.34843	0.34465	0.33156	0.33100	0.33080	**0.33058**
Average MAE	**0.22158**	0.22718	0.22631	0.22557	0.22490	0.22408
Model Sparsity	n/a	n/a	0	**0.9248**	0.8100	0.7504



RMSE improvement over Persistence method

From the Table and boxplot:

- All of the spatio-temporal models significantly outperform the local models.
- LASSO-VAR has highest sparsity but lowest accuracy among sparse models.
- CCSC-VAR model has lowest sparsity
- CCSC-VAR model has lowest average RMSE error for 25 wind farms
- The minimum, maximum and average improvements of CCSC-VAR are highest among these models.

3. **Online sparse and adaptive learning for VAR models**

(Lasso) vector auto regression

Power output depends on previous outputs at the wind farm itself and other wind farms:

$$\mathbf{y}_n = \sum_{l=1}^{L} \mathbf{A}_l \mathbf{y}_{n-l} + \boldsymbol{\epsilon}_n$$

Minimize

$$\sum_{t=1}^{T} \qquad || \sum_{l=1}^{L} (\mathbf{A}_l \mathbf{y}_{n-l}) - \mathbf{y}_n ||_2^2$$

(Lasso) vector auto regression

Power output depends on previous outputs at the wind farm itself and other wind farms:

$$\mathbf{y}_n = \sum_{l=1}^{L} \mathbf{A}_l \mathbf{y}_{n-l} + \boldsymbol{\epsilon}_n$$

Minimize

$$\sum_{t=1}^{T} \qquad || \sum_{l=1}^{L} (\mathbf{A}_l \mathbf{y}_{n-l}) - \mathbf{y}_n ||_2^2$$

(Lasso) vector auto regression

Power output depends on previous outputs at the wind farm itself and other wind farms:

$$\mathbf{y}_n = \sum_{l=1}^{L} \mathbf{A}_l \mathbf{y}_{n-l} + \boldsymbol{\epsilon}_n$$

Minimize

$$\sum_{t=1}^{T} \quad ||\sum_{l=1}^{L} (\mathbf{A}_l \mathbf{y}_{n-l}) - \mathbf{y}_n||_2^2 + \lambda \sum_{l=1}^{L} ||\mathbf{A}_l||$$

- sparse coefficient matrices A_l

(Lasso) vector auto regression

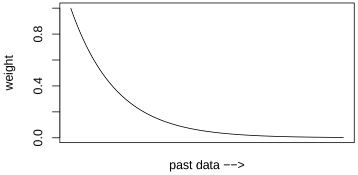
Power output depends on previous outputs at the wind farm itself and other wind farms:

$$\mathbf{y}_n = \sum_{l=1}^{L} \mathbf{A}_l \mathbf{y}_{n-l} + \boldsymbol{\epsilon}_n$$

Minimize

$$\sum_{t=1}^{T} \nu^{N-n} || \sum_{l=1}^{L} (\mathbf{A}_l \mathbf{y}_{n-l}) - \mathbf{y}_n ||_2^2 + \lambda \sum_{l=1}^{L} ||\mathbf{A}_l||$$

- sparse coefficient matrices A_l
- time-adaptive coefficients



VAR Estimation

Cyclic coordinate descent algorithm:
cyclically update coefficients until convergence:

$$A_l[i,j] \leftarrow \frac{\text{sign}(K_N)(|K_N| - \lambda)_+}{L_N}$$

$$K_N = \sum_{n=1}^{N} \nu^{N-n} y_{n-l}[j](y_n[i] - \hat{y}_n[i] + A_l[i,j] y_{n-l}[j])$$

$$L_N = \sum_{n=1}^{N} \nu^{N-n} y_{n-l}[j]^2$$

VAR Estimation

Cyclic coordinate descent algorithm:
cyclically update coefficients until convergence:

$$A_l[i,j] \leftarrow \frac{\mathrm{sign}(K_N)(|K_N| - \lambda)_+}{L_N}$$

$$\begin{aligned}
K_N &= \sum_{n=1}^{N} \nu^{N-n} y_{n-l}[j](y_n[i] - \hat{y}_n[i] + A_l[i,j] y_{n-l}[j]) \\
&= \nu K_{N-1} + y_{N-l}[j](y_N[i] - \hat{y}_N[i] + A_l[i,j] y_{N-l}[j]) \\
L_N &= \sum_{n=1}^{N} \nu^{N-n} y_{n-l}[j]^2 \\
&= \nu L_{N-1} + y_{N-l}[j]^2
\end{aligned}$$

$\rightarrow$ data need not to be stored

VAR Estimation

Cyclic coordinate descent algorithm:
cyclically update coefficients until convergence:

$$A_l[i,j] \leftarrow \frac{\text{sign}(K_N)(|K_N| - \lambda)_+}{L_N}$$

$$\begin{aligned}
K_N &= \sum_{n=1}^{N} \nu^{N-n} y_{n-l}[j](y_n[i] - \hat{y}_n[i] + A_l[i,j]y_{n-l}[j]) \\
&= \nu K_{N-1} + y_{N-l}[j](y_N[i] - \hat{y}_N[i] + A_l[i,j]y_{N-l}[j]) \\
L_N &= \sum_{n=1}^{N} \nu^{N-n} y_{n-l}[j]^2 \\
&= \nu L_{N-1} + y_{N-l}[j]^2
\end{aligned}$$

$\rightarrow$ data need not to be stored

- initialize coordinate descent with previous estimates

$\rightarrow$ fast convergence

Simulation study

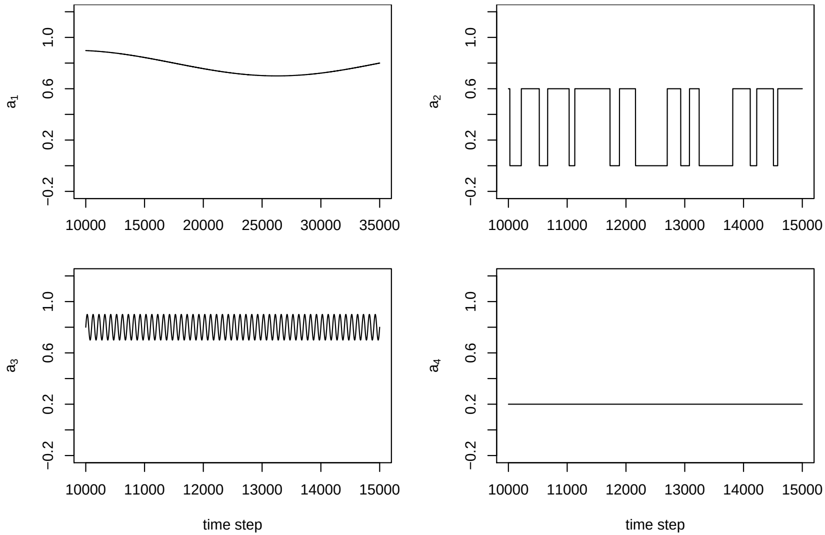
1st-order VAR time-series with coefficient matrix

$$\mathbf{A} = \begin{bmatrix} 0.9 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_2 & 0.9 & 0 & 0 & 0 & 0.2 & 0 & 0 \\ 0.1 & 0 & 0 & 0 & a_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_4 & 0 & 0.9 & 0 & 0 & 0 & -0.1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0 & 0 \\ 0 & 0 & -0.1 & 0 & 0 & 0 & 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 \end{bmatrix}$$

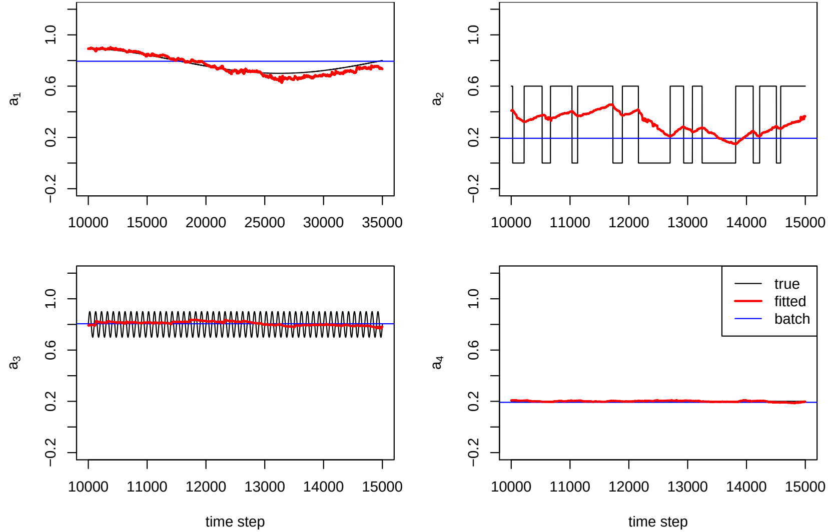
and a white multivariate Gaussian noise.

$\rightarrow$ The interesting aspect is that a_1, a_2, a_3, a_4 are time varying...

Simulation study



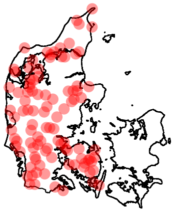
Simulation study

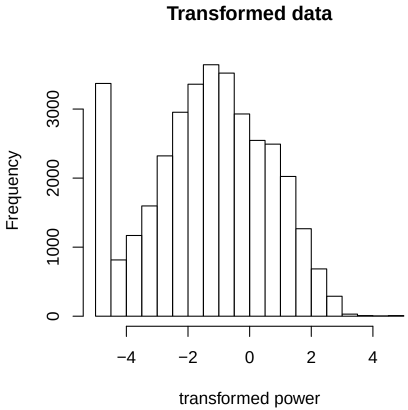


Sparsity: 49% (true: 83%)

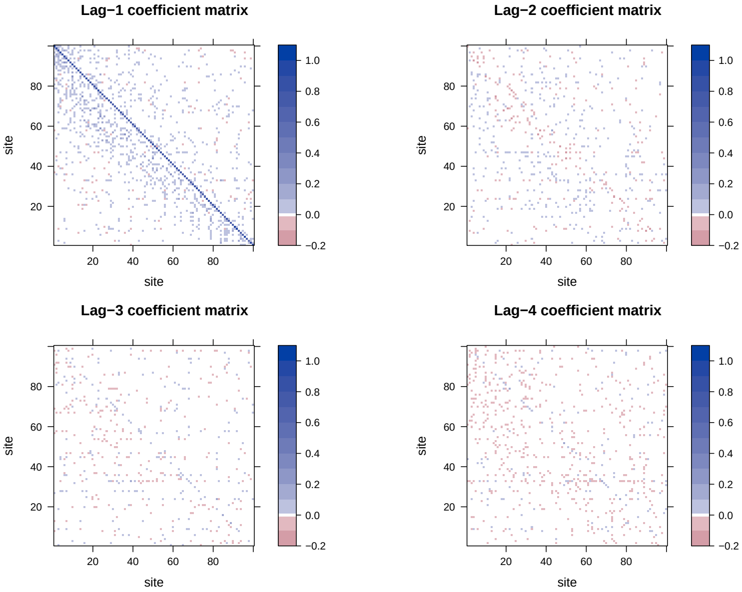
Denmark data

- 100 wind farms (out of 349), 15-min resolution
- logistic transformation
- 2011 (35.036 time steps)
- batch VAR estimation: first 20.000 data
- sorted from West to East



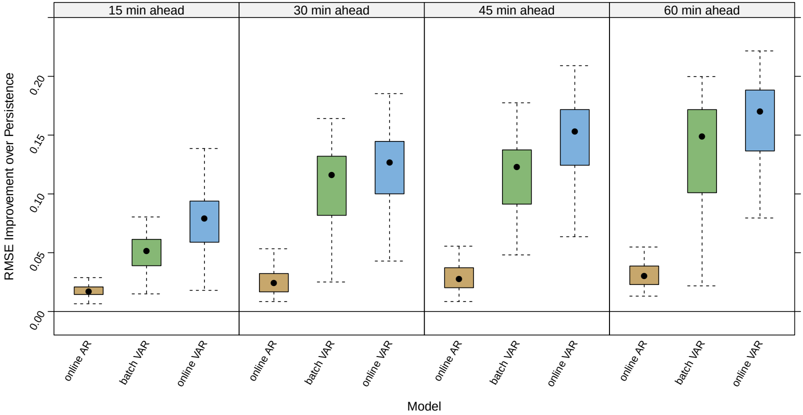


Results



Lag−1 coefficient matrix
Lag−2 coefficient matrix
Lag−3 coefficient matrix
Lag−4 coefficient matrix
site
20
40
60
80
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

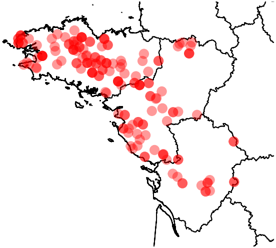
Results

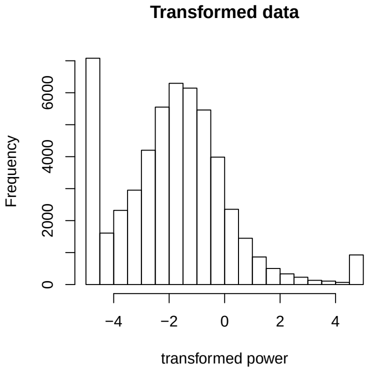


- the VAR model with batch learning outperformed AR models with online learning
- online sparse learning for the VAR model yields substantial extra gains

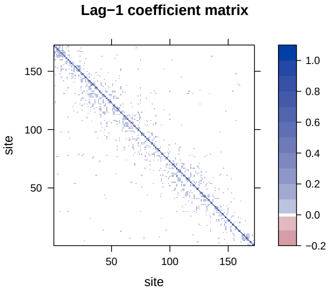
France data

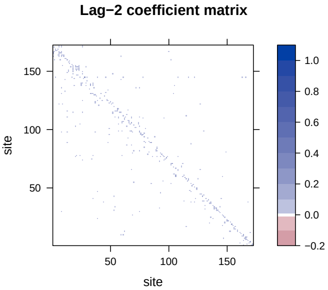
- 172 wind farms, 10-min resolution
- subset 2013 (52.561 time steps)
- logistic transformation
- batch VAR estimation: first 20.000 time steps
- sorted from West to East

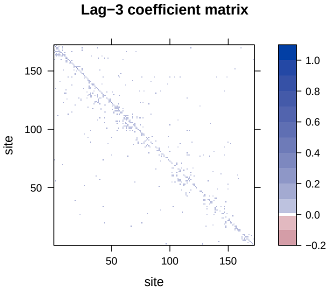




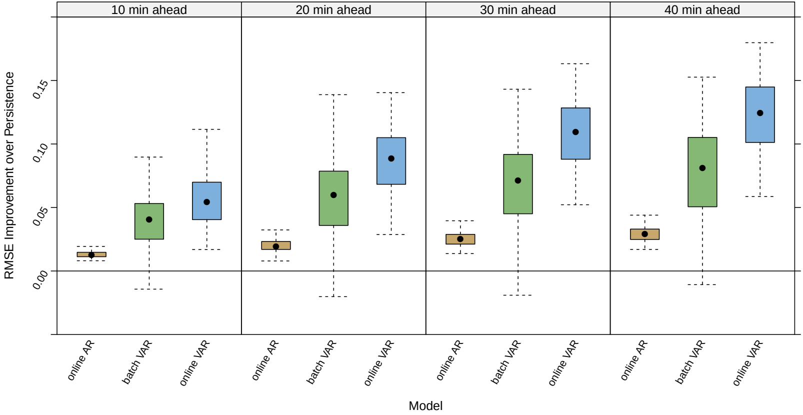
Results


Lag−1 coefficient matrix
site
site
50
100
150
1.0
0.8
0.6
0.4
0.2
0.0
−0.2


Lag−2 coefficient matrix
site
site
50
100
150
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

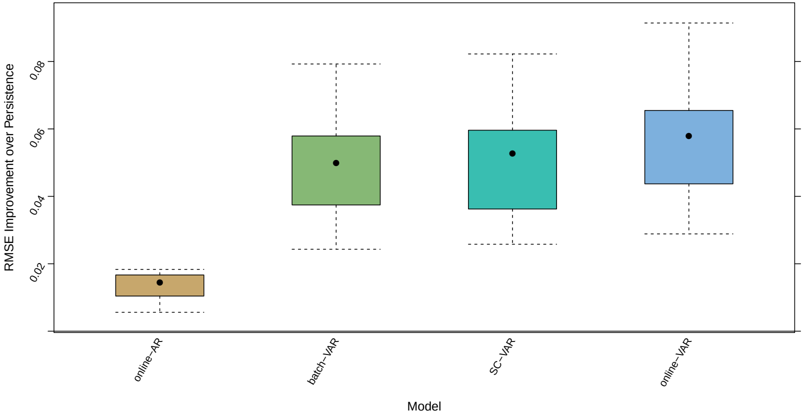

Lag−3 coefficient matrix
site
site
50
100
150
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Results



- the results obtained on the Danish data are confirmed with the French dataset...

Comparison with CCSC-VAR



- the CCSC-VAR outperforms (slightly) the basic VAR with batch learning
- the online sparse VAR estimator does even better

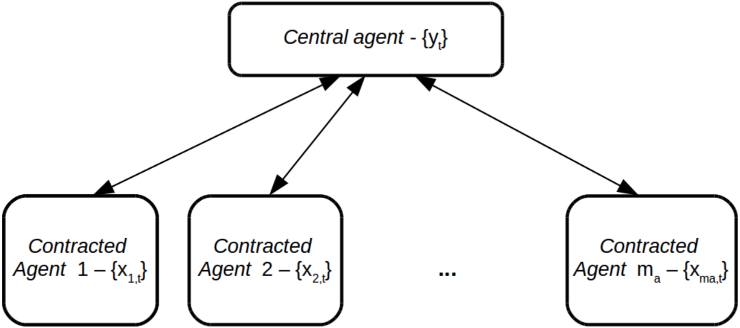
4. **Distributed learning**

Data sharing... or not!

Data sharing... or not!

- To my knowlegde, most players do not want to share their data - even though models and forecasts would highly benefit from that!
- one may design **distributed learning algorithms** that are **privacy-preserving**

- Example setup, with a *central* and *contracted agents*:



- Distributed learning, optimization, etc. is to **play a key role** in future energy analytics

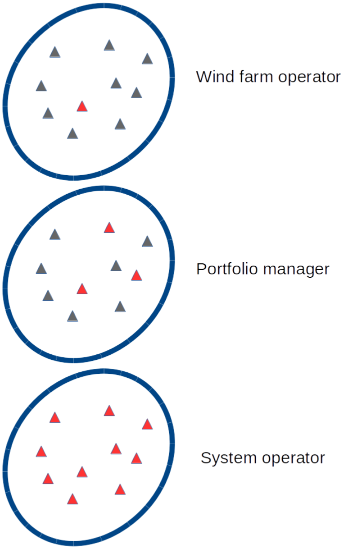
Our mathematical setup

- Wind power generation measurements $x_{j,t}$ are being collected at sites s_j, $j = 1, \ldots, m$ (with t the time index)

- Out of the overall set of wind farms Ω,
  - a **central agent** is interested in a subset of wind farms Ω_p (dim. m_p)
  - **contracted agents** relate to another subset of wind farms Ω_a (dim. m_a)

  Write y_t the wind power production the central agent is interested in predicting

- 3 possible cases in practice:
  - a *wind farm operator* contracting neighbouring wind farms ($m_p = 1$)
  - a *portfolio manager* contracting other wind farms ($m_p > 1$)
  - a *system operator* interested in the aggregate production of all wind farms ($m_p = m$)



AR models with offsite information

- Since restricting ourselves to the very short term, Auto-Regressive (AR) models with offsite information are sufficient

- Such a model reads as

$$y_t = \beta_0 + \sum_{s_j \in \Omega_p} \sum_{\tau=1}^{l} \beta_{j,\tau} x_{j,t-\tau} + \sum_{s_j \in \Omega_a} \sum_{\tau=1}^{l} \beta_{j,\tau} x_{j,t-\tau} + \varepsilon_t$$

  where τ is a lag variable ($\tau = 1, \ldots, l$)

- In a compact form:

$$y_t = \boldsymbol{\beta} \mathbf{x}_t + \varepsilon_t$$

- As the number of coefficients may be large, we use a Lasso-type estimator, i.e.,

$$\hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1$$

- After estimating $\boldsymbol{\beta}$ a forecast is given by

$$\hat{y}_{t+1|t} = \hat{\boldsymbol{\beta}} \mathbf{x}_{t+1}$$

Distributed learning with ADMM

- The **Alternating Direction Method of Multipliers** (ADMM), is a widely used decomposition approach that allows to split a learning problem among features
- The Lasso estimation problem is first reformulated as

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{y} - \mathbf{A}\boldsymbol{\beta}\|_2^2 + \lambda\|\mathbf{z}\|_1 \\ \text{s.t.} \quad & \boldsymbol{\beta} - \mathbf{z} = 0 \end{aligned}$$

- It is then split among agents by setting

$$\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_{m_a+m_p}]$$
$$\mathbf{A} = [\mathbf{A}_1 \; \mathbf{A}_2 \; \ldots \; \mathbf{A}_{m_a+m_p}]$$

- The iterative solving approach is then defined such that, at iteration k,

$$\text{(agent } j) \qquad \boldsymbol{\beta}_j^k = \underset{\boldsymbol{\beta}_j}{\operatorname{argmin}} \left(\|\mathbf{A}_j\boldsymbol{\beta}_j - \mathbf{y}_j^{k-1}\|_2^2 + \frac{2\lambda}{\rho}\|\boldsymbol{\beta}_j\|_1 \right)$$

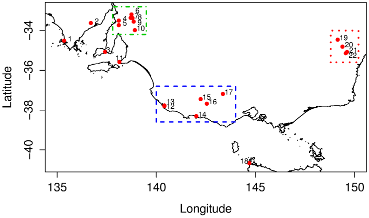
$$\text{(central agent)} \qquad \overline{\mathbf{z}}^k = \frac{1}{(l+1)(m_a+m_p)+\rho} \left(\mathbf{y} + \overline{\mathbf{A}\boldsymbol{\beta}}^k \rho u^{k-1} \right)$$

$$\mathbf{u}^k = \rho u^{k-1} + \overline{\mathbf{A}\boldsymbol{\beta}}^k - \overline{\mathbf{z}}^k$$

(where $\mathbf{y}_j^{k-1} = \mathbf{A}_j\boldsymbol{\beta}_j^{k-1} - \overline{\mathbf{A}\boldsymbol{\beta}}^{k-1} + \overline{\mathbf{z}}^{k-1} - \mathbf{u}^{k-1}$, and $\overline{\mathbf{A}\boldsymbol{\beta}}^k = \sum_{j=1}^{m_a+m_p} \mathbf{A}_j\boldsymbol{\beta}_j$)

Australia

- Data from Australian Electricity Market Operator (AEMO)
- Data is public and shared by Uni. Strathclyde (Jethro Browell) and DTU
- 22 wind farms over a period of 2 years
- 5-minute resolution coarsened to 30 minutes



France

- Data from Enedis (formerly EDF Distribution)
- Data is confidential!
- 187 wind farms over a period of 3 years (only 85 used here)
- 10-minute resolution coarsened to 60 minutes

Only out-of-sample evaluation of genuine 1-step ahead forecasting!

Case 1: Wind farm operator

- Using Australian test case for a simple illustration at a single wind farm
- Comparison of persistence benchmark, local model (AR), and distributed learning model (ARX)

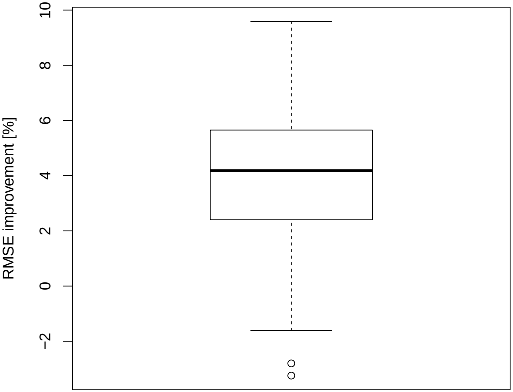
Table: Comparative results for distributed learning (ARX model), as well as persistence and AR benchmarks, at an Australian wind farm (wind farm no. 8) for 30-min ahead forecasting.

	Persistence	AR	ARX (dist. learning)
RMSE [% nom. capacity]	3.60	3.57	3.52
Improvement [%]	-	0.8	2.2

- The improvement is modest, but significant
- This is while the central agent (wind farm 8) never had access to data of contracted wind farms
- Thanks to L_1-penalization, the number of contracted wind farm is very limited

Case 1: Wind farm operator (2)

- Extensive analysis based on the French dataset
- Improvement of distributed learning over local model only, in terms of RMSE



- Improvement is nearly always there
- It ranges from modest to substantial
- This obviously depends on the wind farm location

Case 2: Portfolio manager

- Using French test case
- We randomly pick 8 wind farms to build a portfolio
- Comparison of persistence benchmark, local model (AR), and distributed learning model (ARX)

Table: Comparative results for distributed learning (ARX model), as well as persistence and AR benchmarks, for a portfolio of 8 wind farms of the French dataset (randomly chosen) for 1-hour ahead forecasting.

	Persistence	AR	ARX (dist. learning)
RMSE [% nom. capacity]	3.99	3.67	3.38
Improvement [%]	-	8.2	15.3

- The improvement is substantial
- Again, thanks to L_1-penalization, the number of contracted wind farm is very limited
- Simulation studies may allow to look at how improvement relates to portfolio size, wind farm distribution, etc.

Case 3: System operator

- Using French test case
- The system operator aims to predict the aggregate of all wind farms, though never accessing the wind farm data(!)
- Comparison of persistence benchmark, local model (AR), and distributed learning model (ARX)

Table: Comparative results for distributed learning (ARX model), as well as persistence and AR benchmarks, for the aggregate of all 85 French wind farms for 1-hour ahead forecasting.

	Persistence	AR	ARX (dist. learning)
RMSE [% nom. capacity]	2.88	2.10	2.05
Improvement [%]	-	27.1	28.8

- The improvement is modest, since an AR model obviously does very well for aggregated wind power production
- Though, the practical interest is huge, since data does not need to eb exchanged
- More complex models (e.g., regime-switching) may yield higher improvements

Concluding thoughts

- **High-dimensional** and **distributed learning** have a **bright future in energy analytics**, since
  - high quantity of distributed data is being collected
  - data-driven and expert input to reveal and maintain sparsity
  - most actors do not want to share their data (unless forced to do so)

- Some interesting **future developments**:
  - online distributed learning (i.e., merger of ideas persented), to lighten computational costs and exchange/communication needs
  - broaden the applicability to a wide class of models, e.g., with regime switching and regression on input weather forecasts
  - design distributed computation and data sharing markets!

Thanks for your attention!