ECE 598 Computational Power Systems

Renewable Power Forecasting

Vladimir Dvorkin

University of Michigan

Course overview

Covered topics:

1. Intro to computational power systems
2. Duality, optimality conditions, and electricity pricing
3. Optimal power flow & Locational pricing
4. Distributed/decentralized optimization (ADMM)
5. ADMM applications to optimal power flow
6. Online feedback optimization (OFO)
7. OFO application to Volt/VAr control in distribution grid
8. OFO application to real-time economic re-dispatch
9. **Today:** Renewable power forecasting

What comes up next?

- Advanced regression analysis
- Decision-focused analytics
- Final project presentation (**April 18th**)

Why forecasting?

- Forecasting is the first step in decision-making
- Resolves (some) uncertainty of decision-making inputs
- Brings confidence to your decision-making

Need for forecasting in power systems

Who needs a forecast?

- Power producers: conventional generators, wind farm operators
- Utility companies, large industrial consumers, aggregators, etc.
- Independent system operators (US), system and market operators (Europe)

What to forecast?

- Renewable power
- Electricity demand
- Day-ahead and real-time prices
- Real-time system imbalance/congestion
- Any other information relevant to your decision-making...

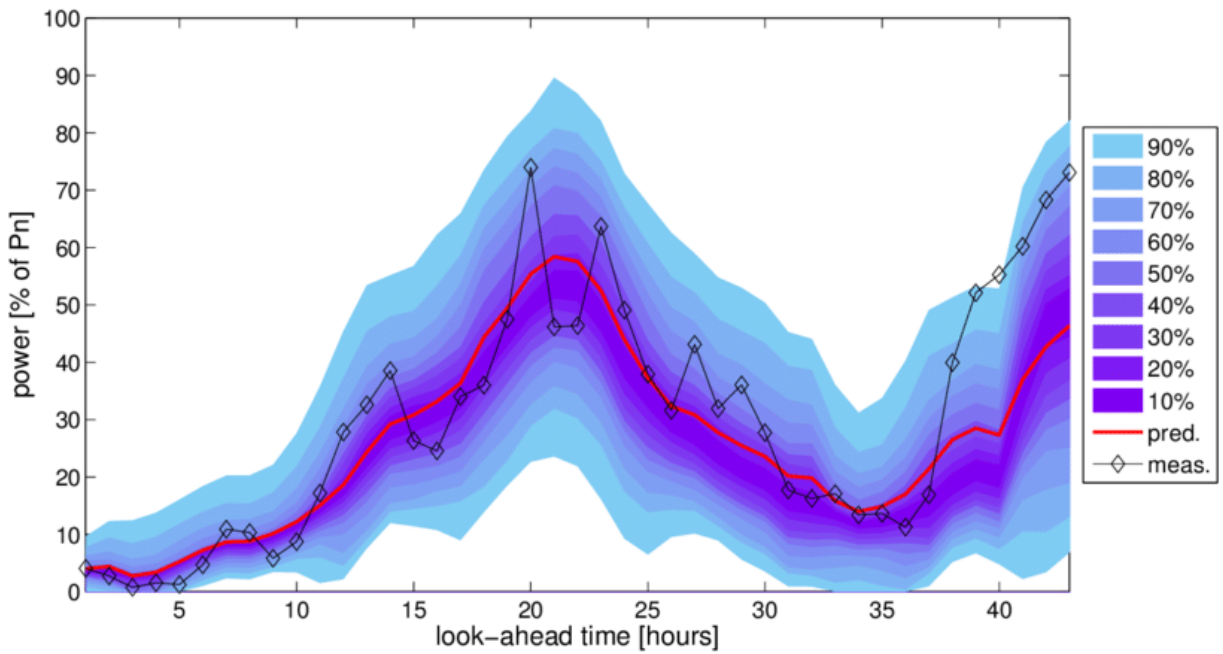
Renewable energy forecasts in decision-making

Forecast provides inputs to many decision-making problems:

- Reserve quantification (i.e., backup capacity for the system operator)
- Unit commitment, economic dispatch, contingency screeining, etc.
- Trading strategy for renewables, aggregators, utilities, etc.

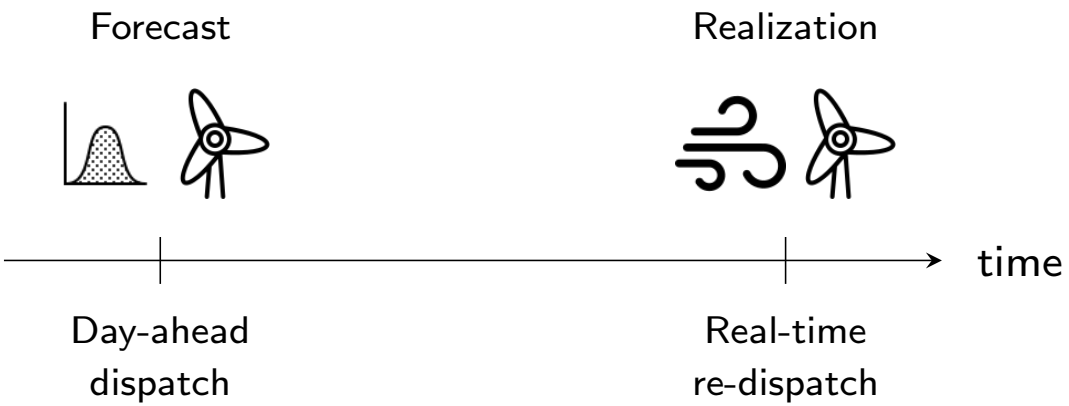
Relevant inputs include:

- Deterministic forecasts
- Probabilistic forecasts as quantiles, intervals, and predictive distributions
- Probabilistic forecasts in the form of trajectories/scenarios



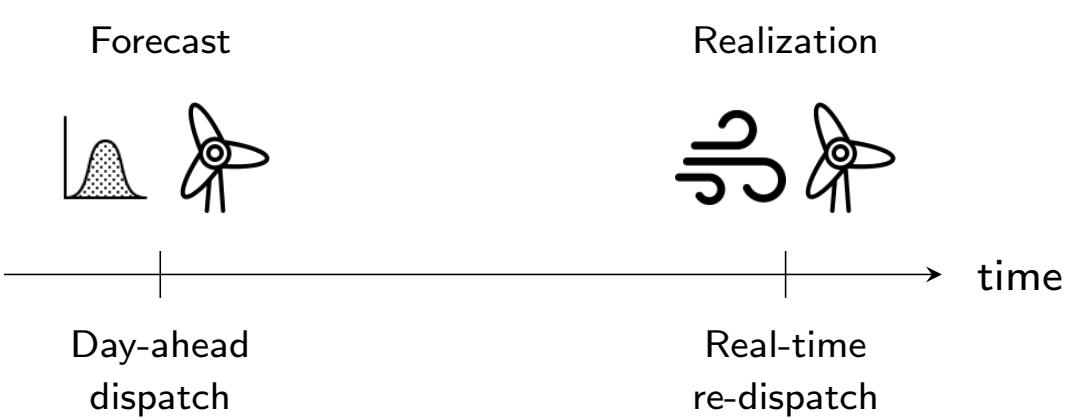
Pinson et al. "Evaluation of nonparametric probabilistic forecasts of wind power"

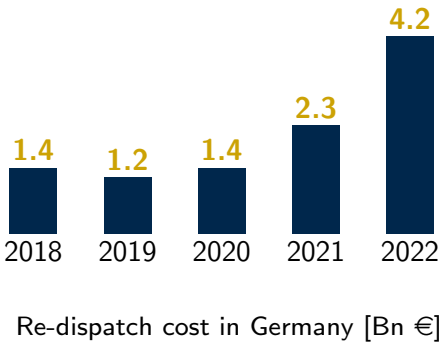
Forecasting for power system dispatch



- Two-stage decision-making to manage uncertainty of renewables:
  - Day-ahead: minimize the cost of power supply using forecast
  - Real-time: costly re-dispatch to accommodate forecast errors
- The costs of real-time re-dispatch increase in renewable power capacity
- Electricity prices (and thus revenues) are a function of the forecast

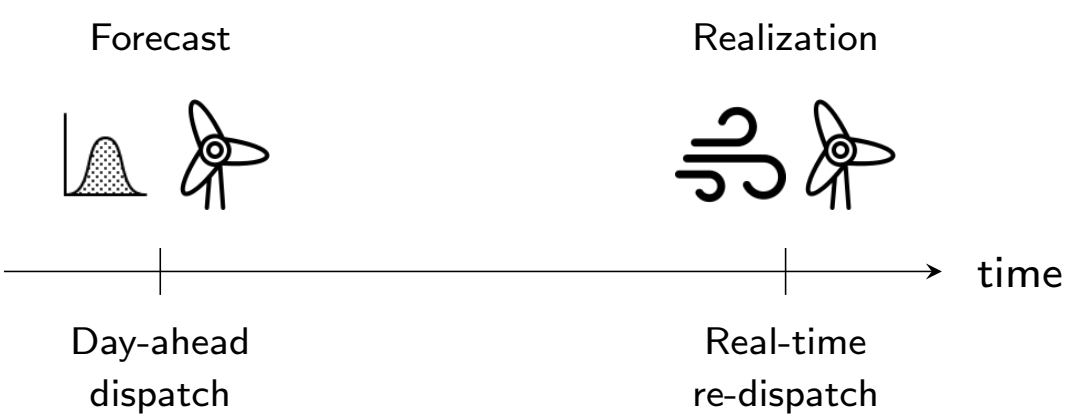
Forecasting for power system dispatch

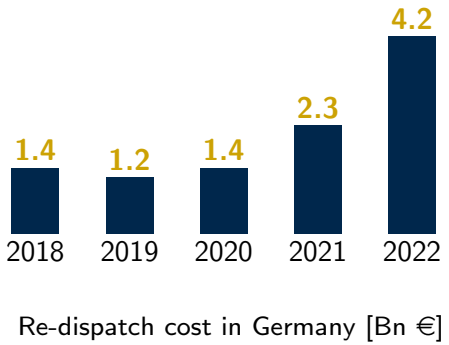




- Two-stage decision-making to manage uncertainty of renewables:
  - Day-ahead: minimize the cost of power supply using forecast
  - Real-time: costly re-dispatch to accommodate forecast errors
- The costs of real-time re-dispatch increase in renewable power capacity
- Electricity prices (and thus revenues) are a function of the forecast

Forecasting for power system dispatch





- Two-stage decision-making to manage uncertainty of renewables:
  - Day-ahead: minimize the cost of power supply using forecast
  - Real-time: costly re-dispatch to accommodate forecast errors
- The costs of real-time re-dispatch increase in renewable power capacity
- Electricity prices (and thus revenues) are a function of the forecast

Need for a "good" power forecast to reduce the cost of uncertainty

Example of market outcomes

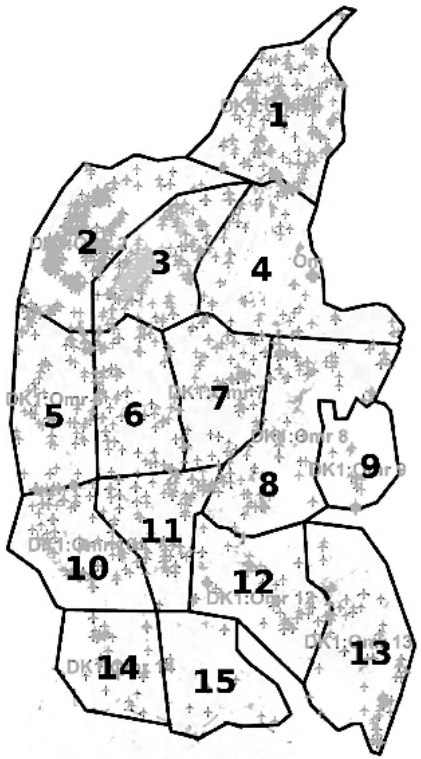
- Dutch electricity pool simulated over a year
- 15-MW wind farm with no storage and control over output
- Point and probabilistic forecasts generated with state-of-the-art tools
- Three forecast models versus perfect predictions
- Market outcomes are very sensitivity to a forecast model
- Which forecast model do you chose?

	Pers.	**Adv. point pred.**	**Prob. pred.**	**Perfect pred.**
Contracted energy (GWh)	44.37	45.49	62.37	46.41
Surplus (GWh)	18.12	9.87	4.89	0
Shortage (GWh)	16.08	8.95	20.85	0
Down-regulation costs (10^3 €)	195.72	119.99	42.61	0
Up-regulation costs (10^3 €)	79.59	52.01	61.46	0
Total revenue (10^3 €)	1041.38	1145.69	1212.61	1317.69
Av. down-reg. unit cost (€/MWh)	10.80	12.15	8.71	0
Av. up-reg. unit cost (€/MWh)	4.95	5.81	2.95	0
Av. reg. unit cost (€/MWh)	8.05	9.13	4.04	0
Av. energy price (€/MWh)	**22.44**	**24.68**	**26.13**	**28.37**
Part of imbalance (% prod. energy)	**73.69**	**40.55**	**55.46**	**0**
Performance ratio (%)	**79.1**	**86.99**	**92.1**	**100**

Pinson, Chevallier, Kariniotakis. "Trading wind generation from short-term probabilistic forecasts of wind power". IEEE TPWRS

Forecast products

Wind power in Denmark

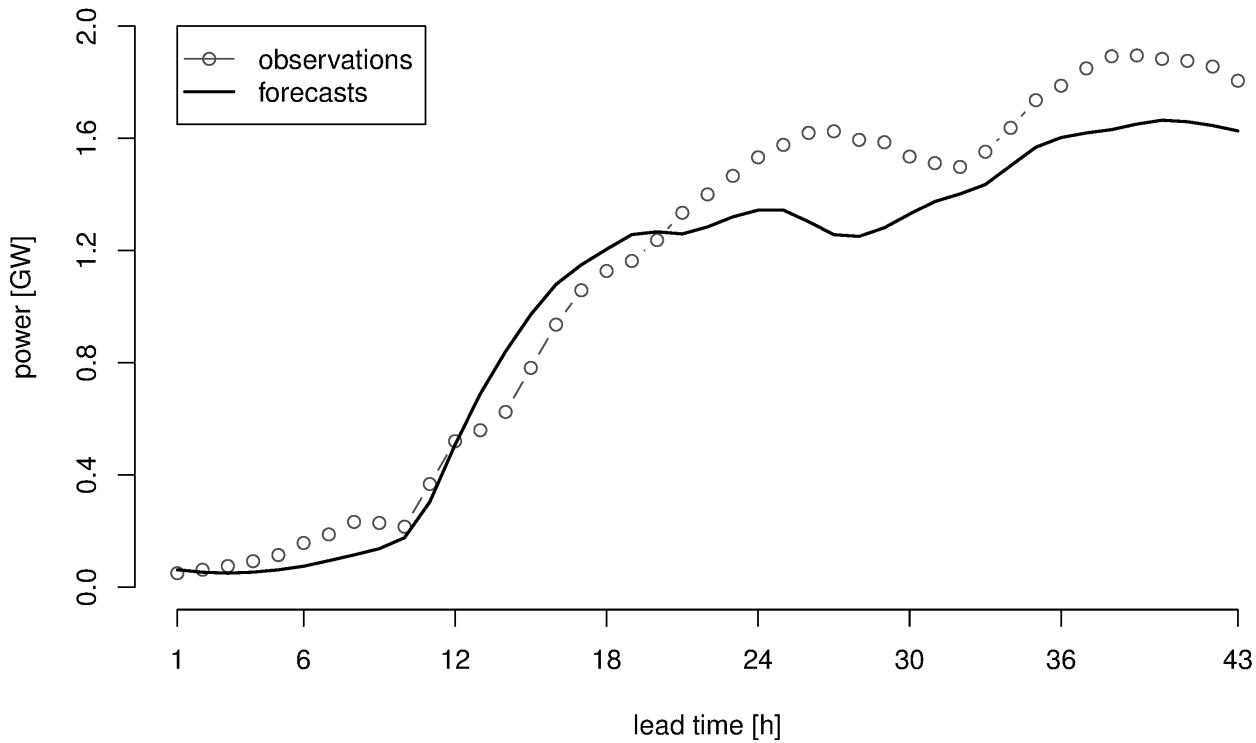


Agg. zone	Orig. zones	% of capacity
1	1, 2, 3	31
2	5, 6, 7	18
3	4, 8, 9	17
4	10, 11, 14, 15	23
5	12, 13	10

Figure: The Western Denmark dataset: original locations for which measurements are available, 15 control zones defined by Energinet, as well as the 5 aggregated zones, for a nominal capacity of around 2.5 GW.

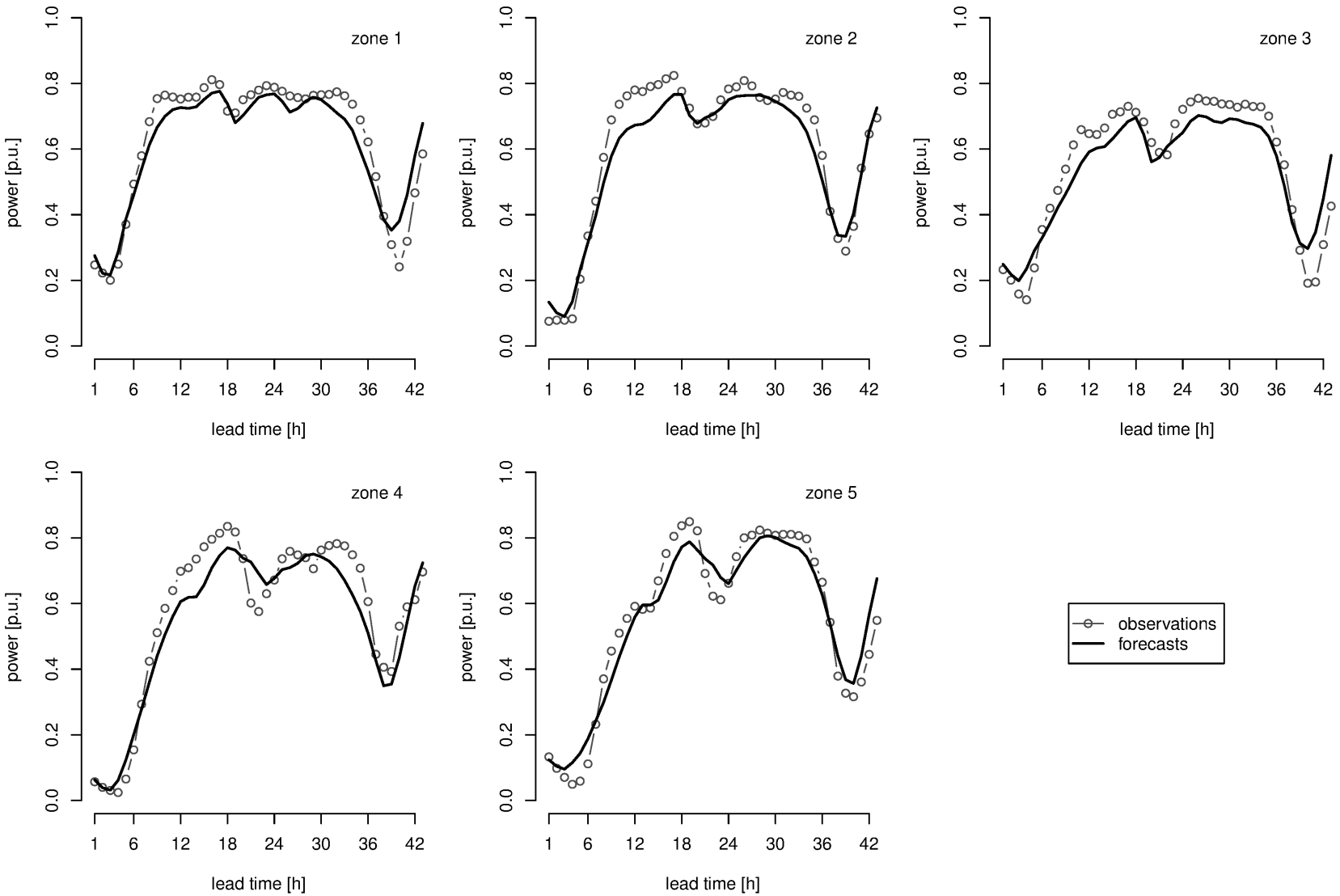
Forecast products – Point forecast

A point forecast informs of the conditional expectation of power generation



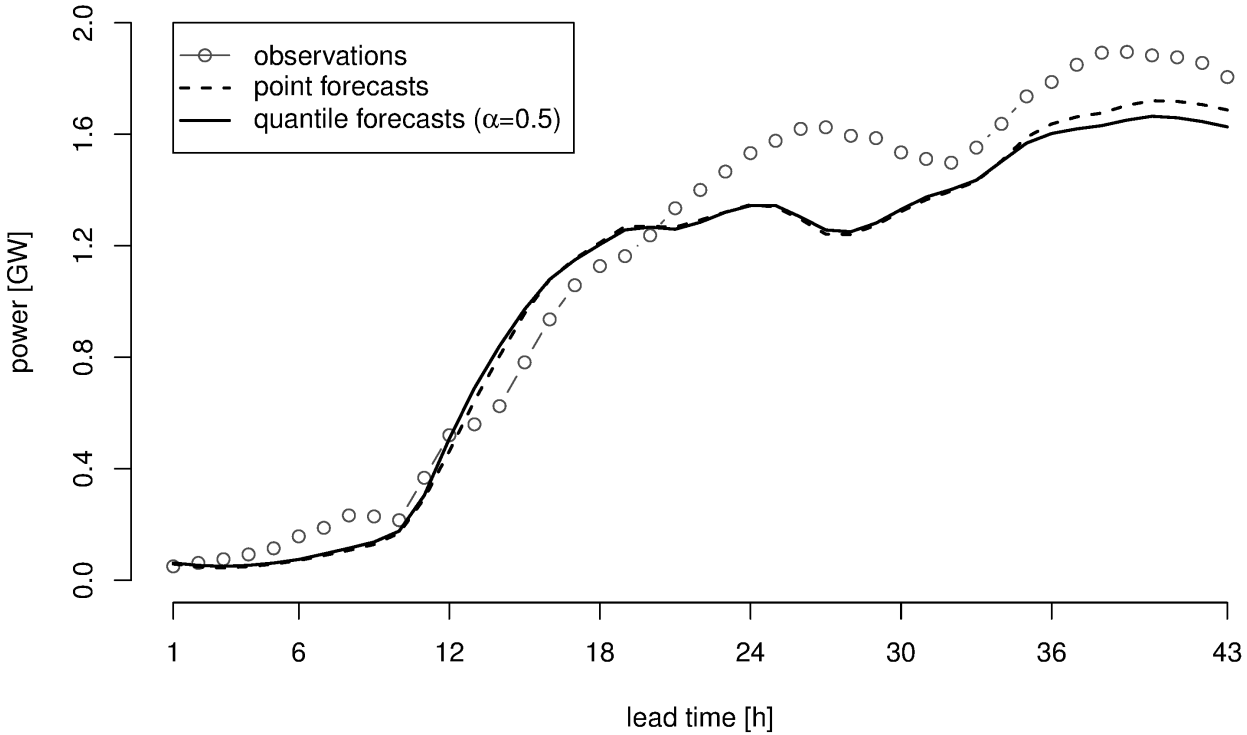
Forecast products – Point forecast

A point forecast informs of the conditional expectation of power generation



Forecast products – Quantile forecast

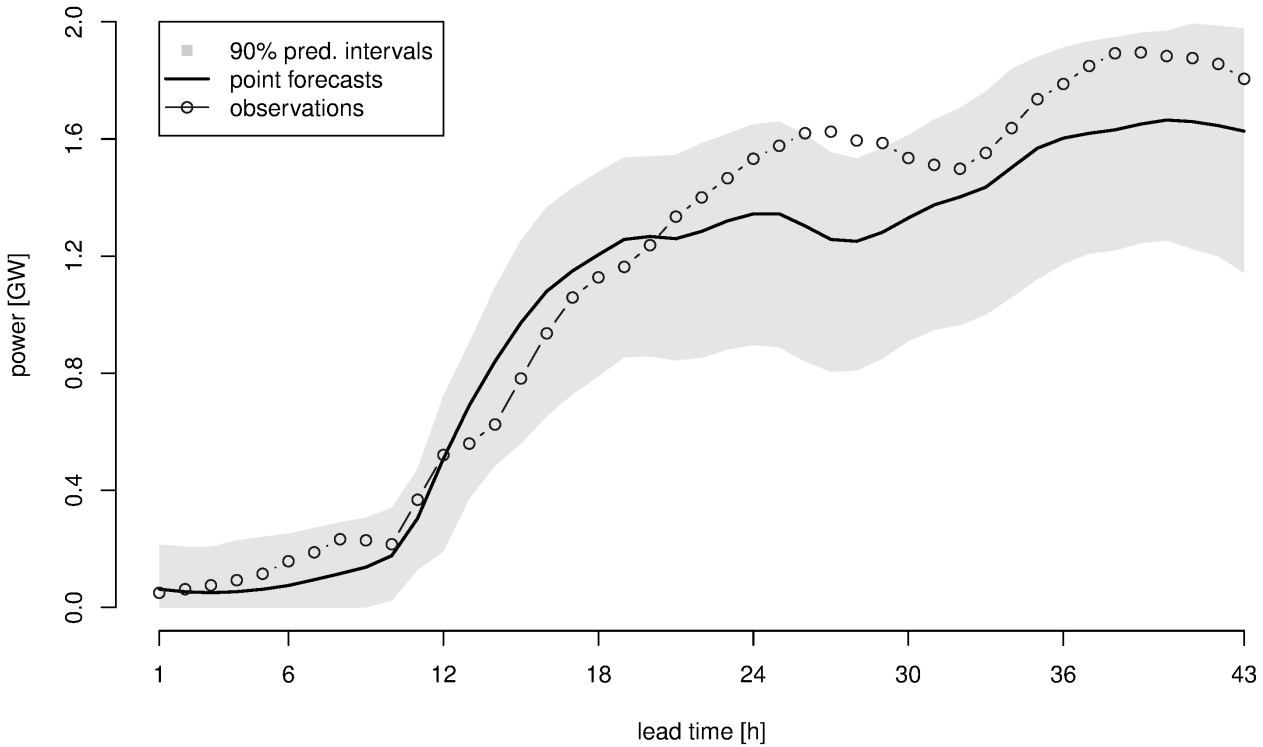
A quantile forecast is to be seen as a probabilistic threshold for power generation



α is the quantile of the cumulative distribution function of the random process

Forecast products – Interval forecast

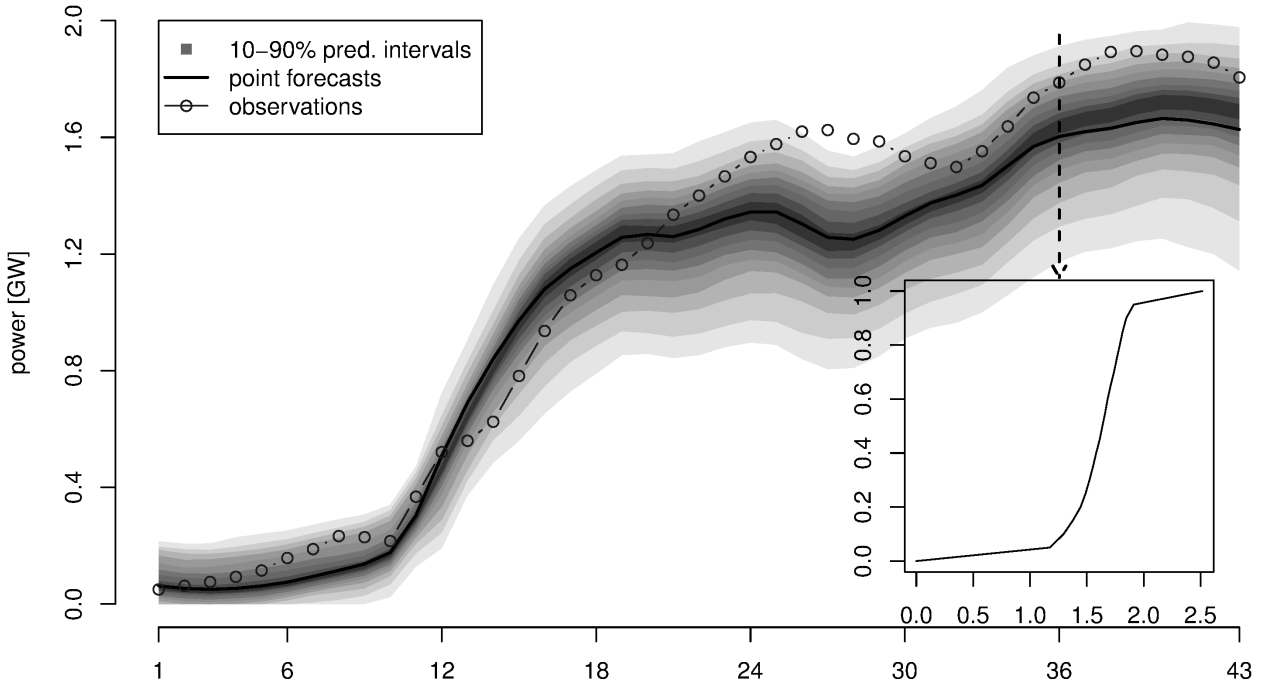
A prediction interval is an interval within which power generation may lie, with a certain probability



Interval here is between 0.05th and 0.95th quantiles (90% probability)

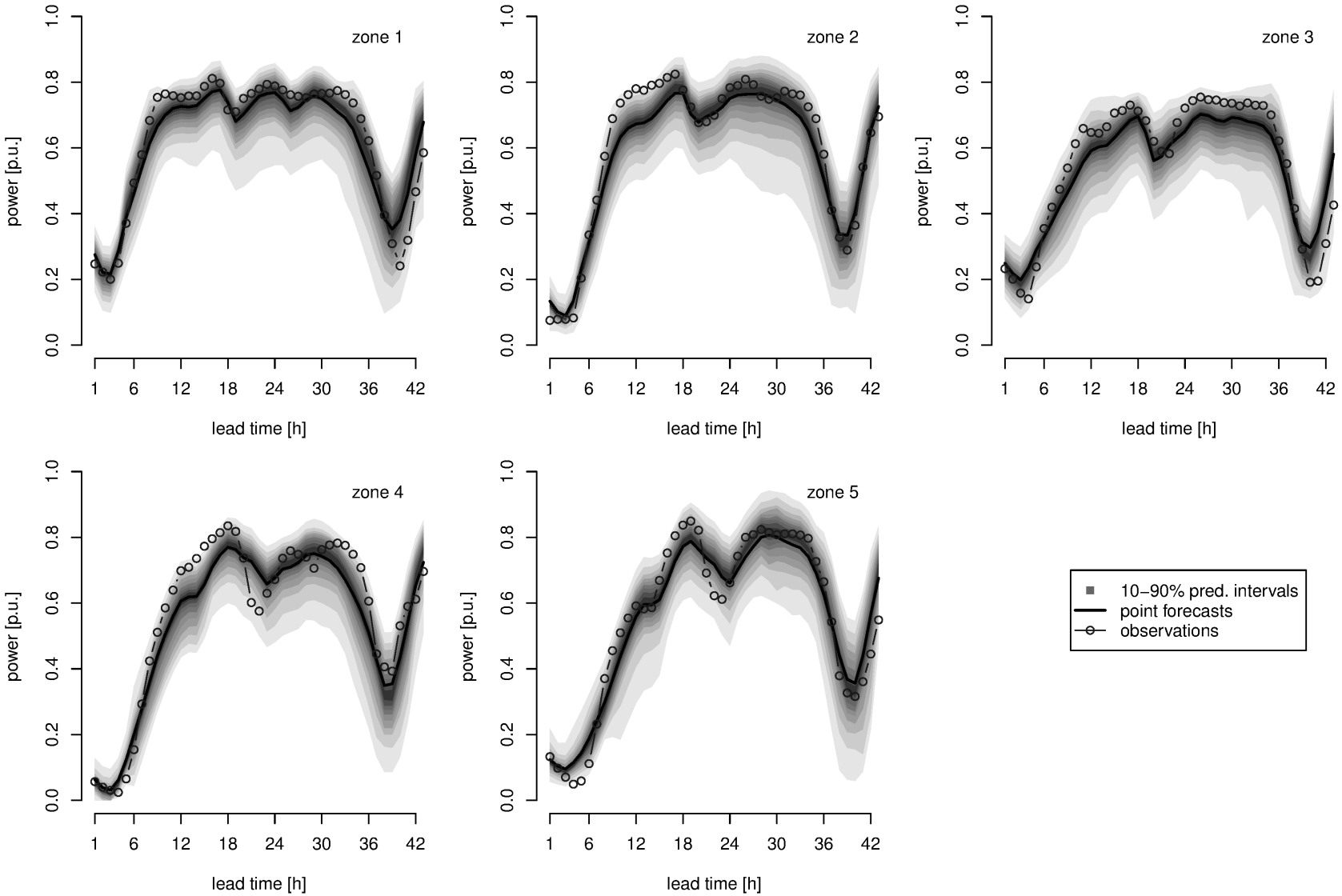
Forecast products – Predictive density

A predictive density fully describes the probabilistic distribution of power generation for every lead time



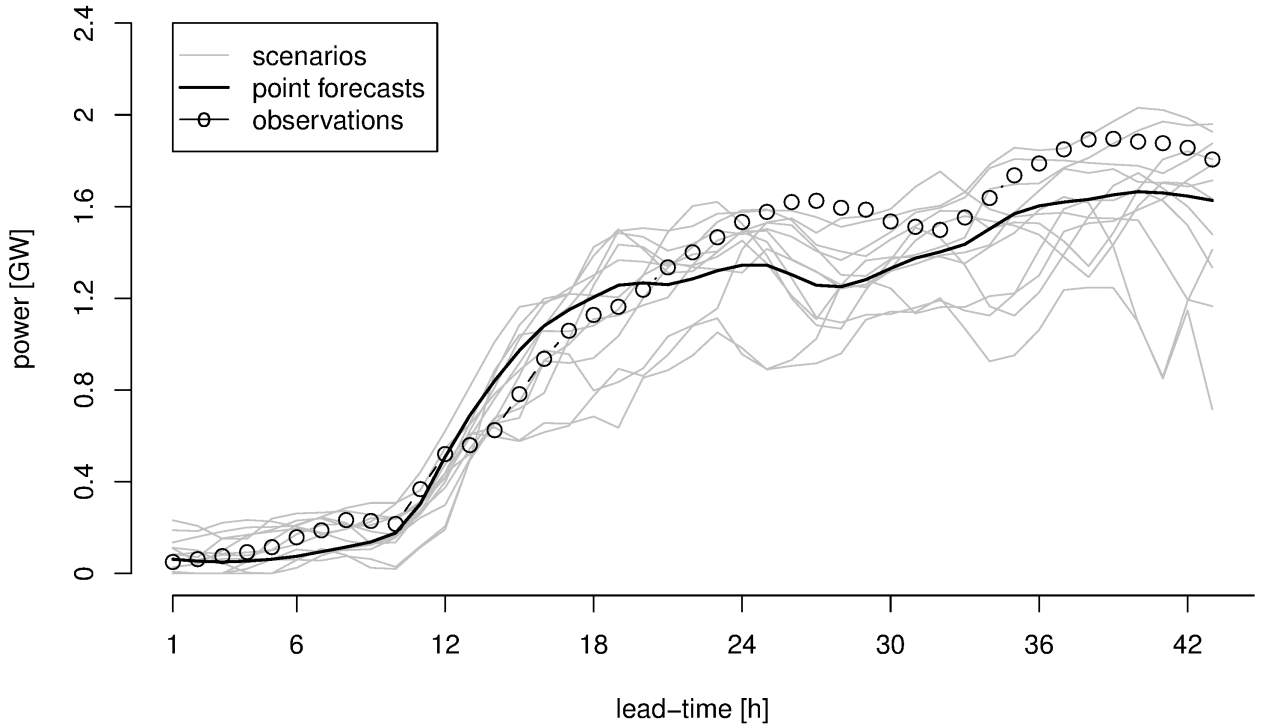
Forecast products – Predictive density

A predictive density fully describes the probabilistic distribution of power generation for every lead time



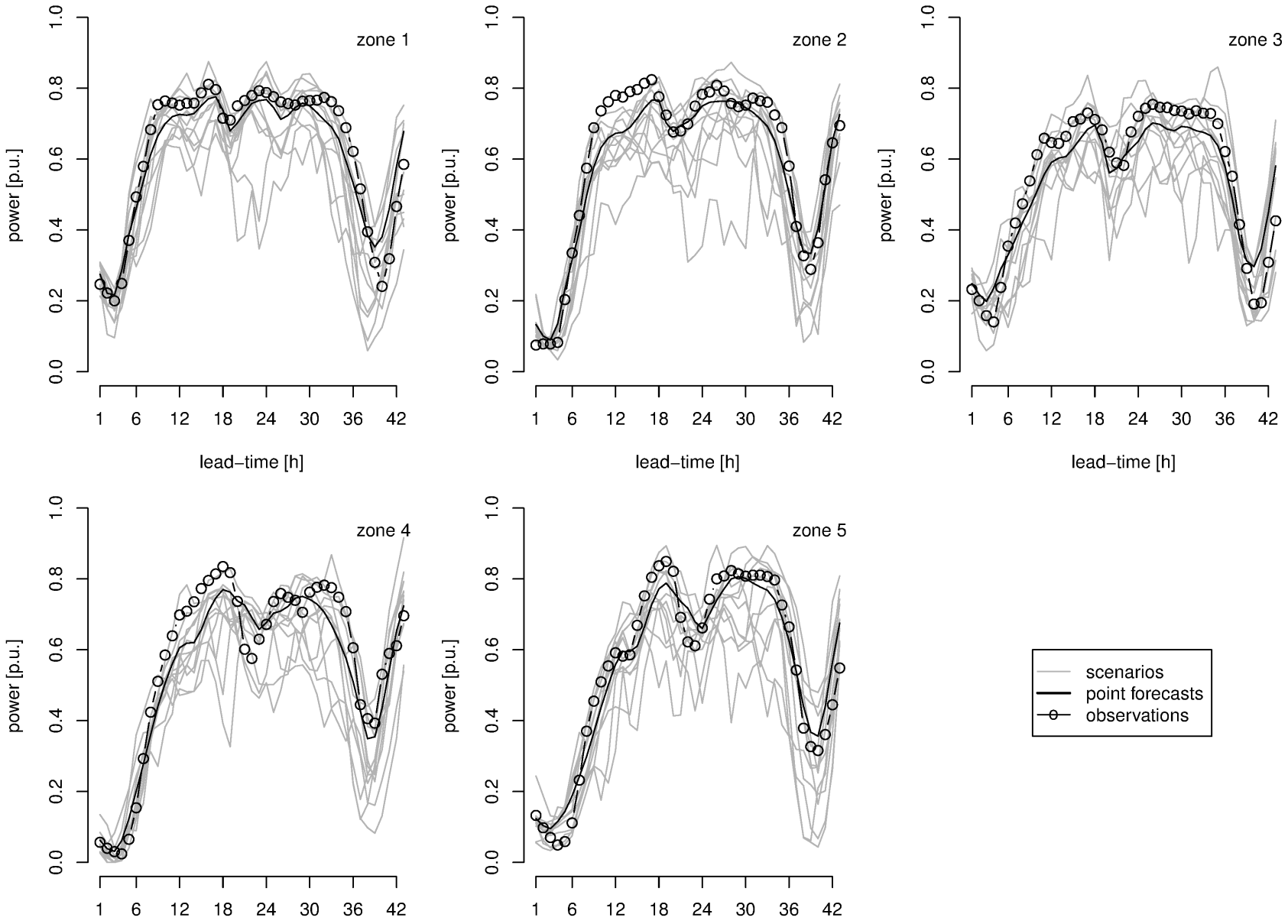
Forecast products – Trajectory (scenarios)

Trajectories are equally-likely samples of multivariate predictive densities for power generation (in time and/or space)



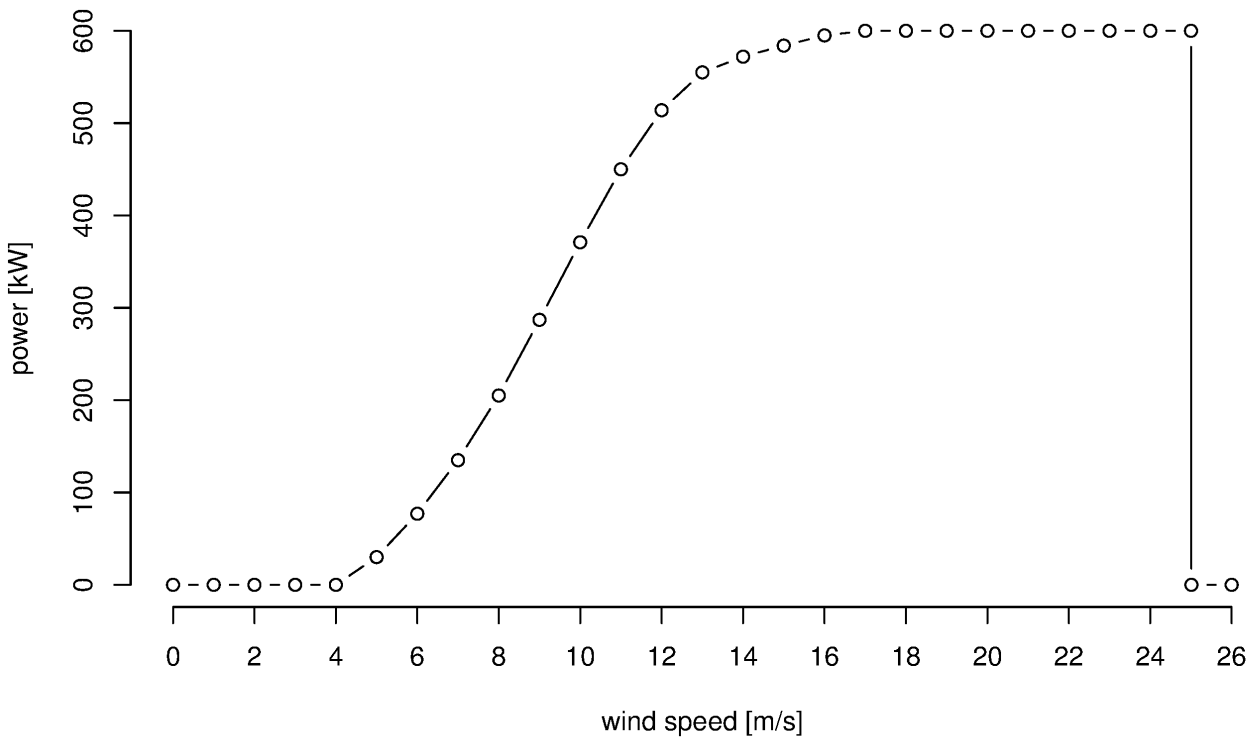
Forecast products – Trajectory (scenarios)

Trajectories are equally-likely samples of multivariate predictive densities for power generation (in time and/or space)



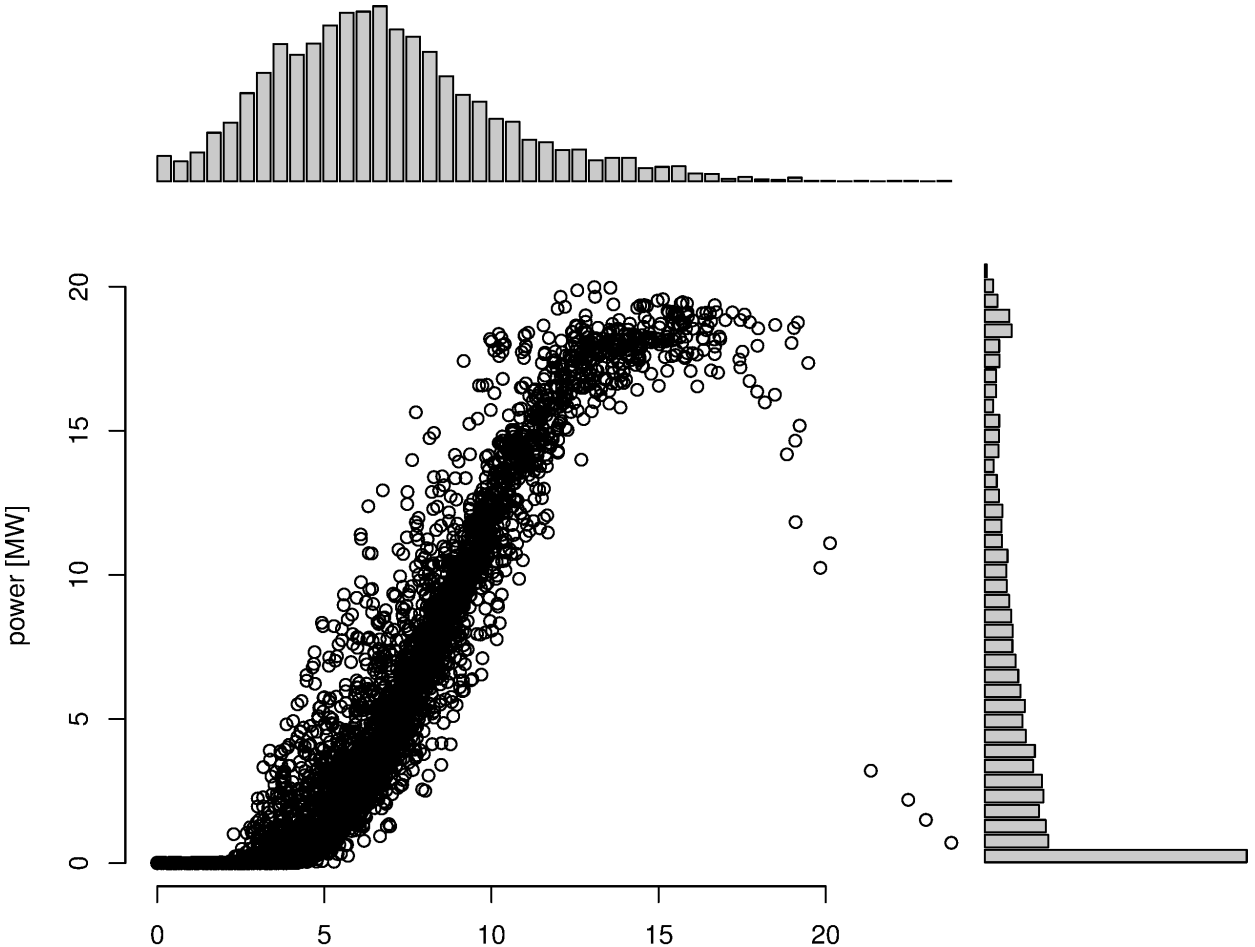
Basics of forecasting

Theoretical wind power curve



- Maps meteorological features to the wind power output
- Dead band and cut-off wind speed
- Easy to scale one turbine to the entire farm

Practical wind power curve



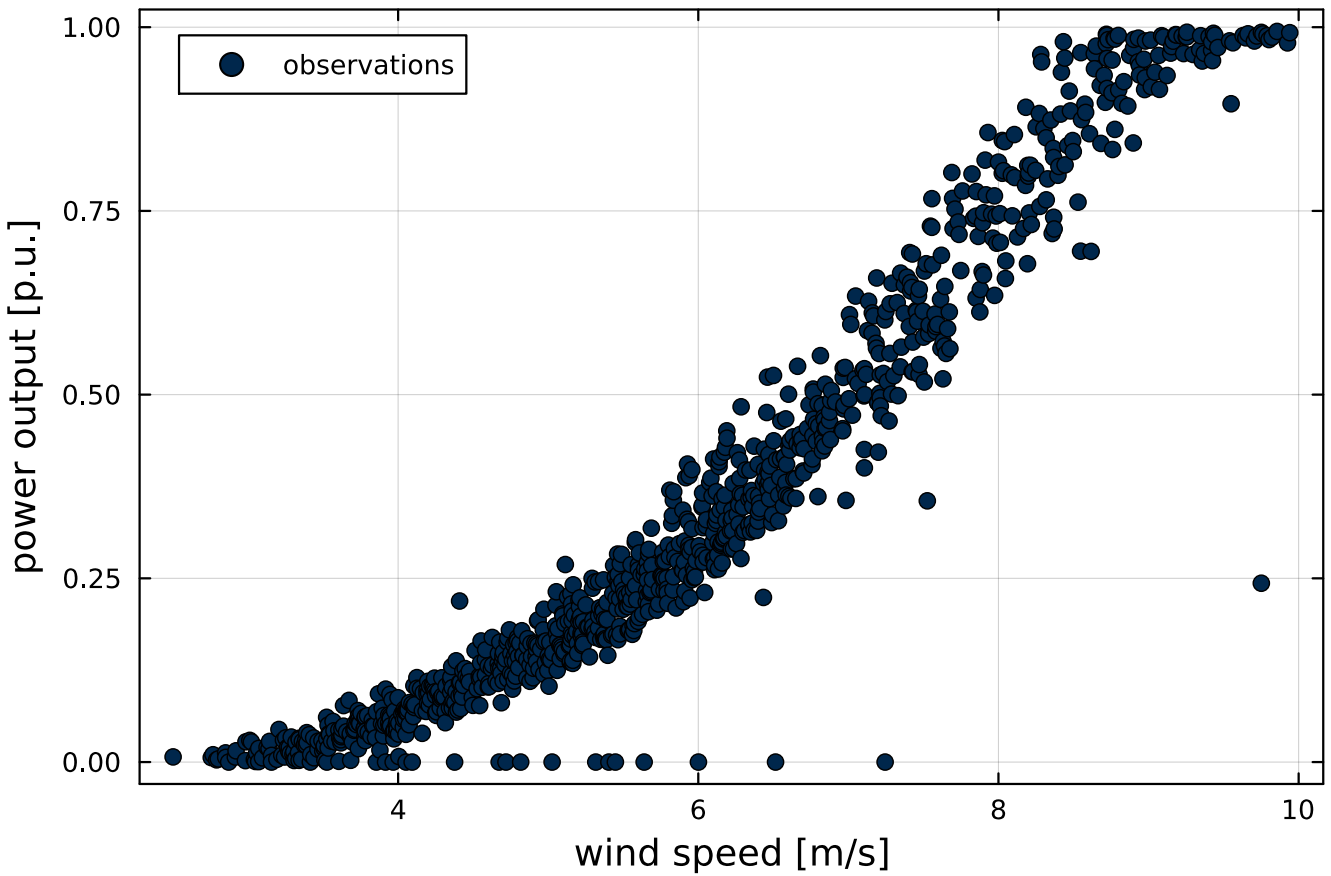
- Actual wind power curve is very different
- A lot of uncertainty in the conversion process
- The uncertainty is amplified by the weather predictions errors

Linear regression for wind power curve fitting

- Dataset $\{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_i, t_i), \ldots, (\mathbf{x}_n, t_n)\}$ of n observations
- Vector x_i of weather features and target t_i for power output
- **Goal:** Fit a linear model to relate power output to weather

Linear regression for wind power curve fitting

- Dataset $\{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_i, t_i), \ldots, (\mathbf{x}_n, t_n)\}$ of n observations
- Vector x_i of weather features and target t_i for power output
- **Goal:** Fit a linear model to relate power output to weather



Linear regression

- Using dataset $\{(x_i, t_i)\}_{i=1}^n$, with feature x and target t, we estimate the model

$$y = \mathbf{w}^\top \mathbf{x} + b$$

  with weight $\mathbf{w}$ and bias b (intercept)

- Loss function measures per-instance loss

$$\mathcal{L}(y, t) = \frac{1}{2}(y - t)^2$$

- Cost function measures the loss across the entire dataset

$$\mathcal{C}(\mathbf{y}, \mathbf{t}) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - t_i)^2$$

- Optimize the model by solving the following convex optimization

$$\begin{aligned} \underset{\mathbf{w}, b}{\text{minimize}} \quad & \frac{1}{2n} \sum_{i=1}^{n} (y_i - t_i)^2 \\ \text{where} \quad & y_i = \mathbf{w}^\top x_i + b \quad \forall i = 1, \ldots, n \end{aligned}$$

- Minimize the cost function in scalar variables w and b

$$\underset{w,b}{\text{minimize}} \quad \frac{1}{2n}\sum_{i=1}^{n}(w^\top x_i + b - t_i)^2$$

What is the natural algorithm to solve this problem?

- Minimize the cost function in scalar variables w and b

$$\underset{w,b}{\text{minimize}} \quad \frac{1}{2n}\sum_{i=1}^{n}(w^\top x_i + b - t_i)^2$$

What is the natural algorithm to solve this problem?

Closed-form solution

- Vectorized features:

$$\mathbf{X} = \begin{bmatrix}\mathbf{1} & \mathbf{x}\end{bmatrix}$$

- Compute the weight and bias:

$$\begin{bmatrix} b \\ w \end{bmatrix} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{t}$$

- Minimize the cost function in scalar variables w and b

$$\underset{w,b}{\text{minimize}} \quad \frac{1}{2n}\sum_{i=1}^{n}(w^\top x_i + b - t_i)^2$$

What is the natural algorithm to solve this problem?

Gradient descent (GD)

- Derivatives of the cost function:

$$\frac{\partial \mathcal{C}}{\partial w} = \frac{\partial}{\partial w}\frac{1}{2n}\sum_{i=1}^{n}(y_i - t_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - t_i)x_i$$

$$\frac{\partial \mathcal{C}}{\partial b} = \frac{\partial}{\partial b}\frac{1}{2n}\sum_{i=1}^{n}(y_i - t_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - t_i)$$

- For some chosen step size $\alpha > 0$ and `iter_max`:

for $k = 1, \ldots,$ `iter_max` **do**

$w \leftarrow w - \alpha\frac{\partial \mathcal{C}}{\partial w}$ weight update

$b \leftarrow b - \alpha\frac{\partial \mathcal{C}}{\partial b}$ bias update

end for

- Minimize the cost function in scalar variables w and b

$$\underset{w,b}{\text{minimize}} \quad \frac{1}{2n}\sum_{i=1}^{n}(w^\top x_i + b - t_i)^2$$

What is the natural algorithm to solve this problem?

Gradient descent (GD)

- Derivatives of the cost function:

$$\frac{\partial \mathcal{C}}{\partial w} = \frac{\partial}{\partial w}\frac{1}{2n}\sum_{i=1}^{n}(y_i - t_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - t_i)x_i$$

$$\frac{\partial \mathcal{C}}{\partial b} = \frac{\partial}{\partial b}\frac{1}{2n}\sum_{i=1}^{n}(y_i - t_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - t_i)$$

- For some chosen step size $\alpha > 0$ and `iter_max`:

for $k = 1, \ldots,$ `iter_max` **do**

$w \leftarrow w - \alpha\frac{\partial \mathcal{C}}{\partial w}$ weight update

$b \leftarrow b - \alpha\frac{\partial \mathcal{C}}{\partial b}$ bias update

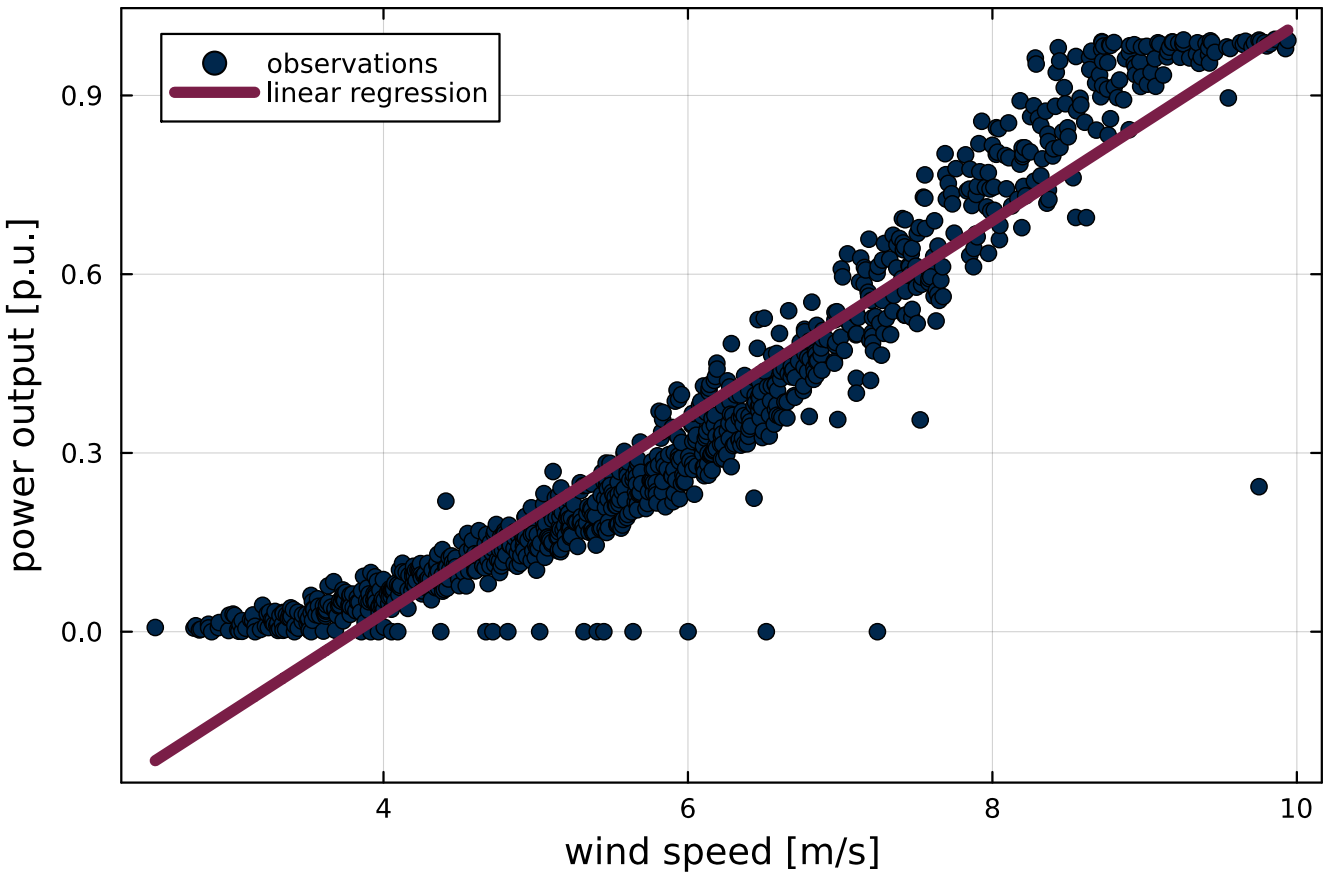
end for

Why GD and not the closed form?

- Applies to a much broader set of models
- In high dimension, matrix inversion is a very expensive operation

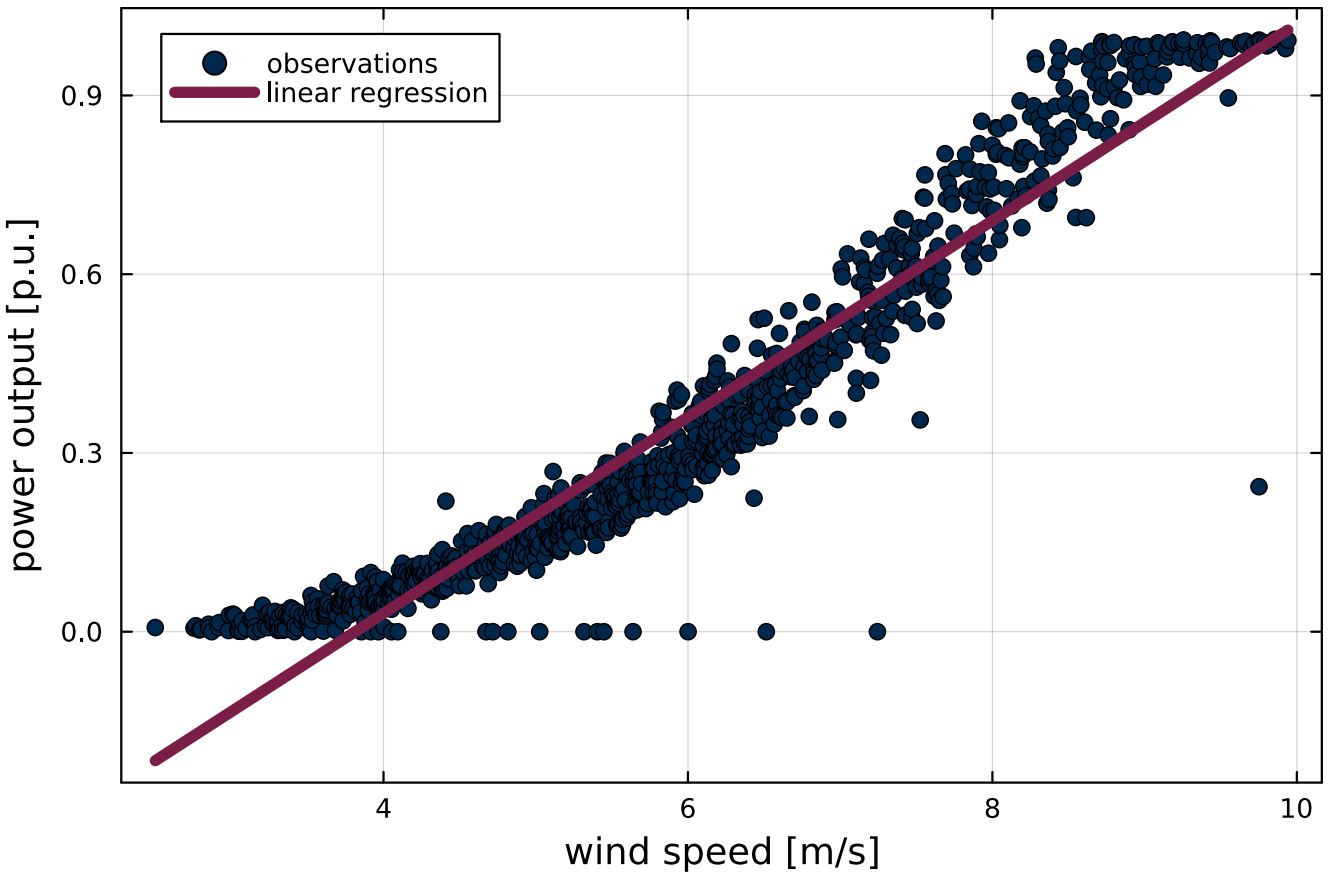
Linear regression results

- For $\alpha = 0.01$, `iter_max`$= 25,000$ and intial model parameters $(w, b) = (0, 0)$:



Linear regression results

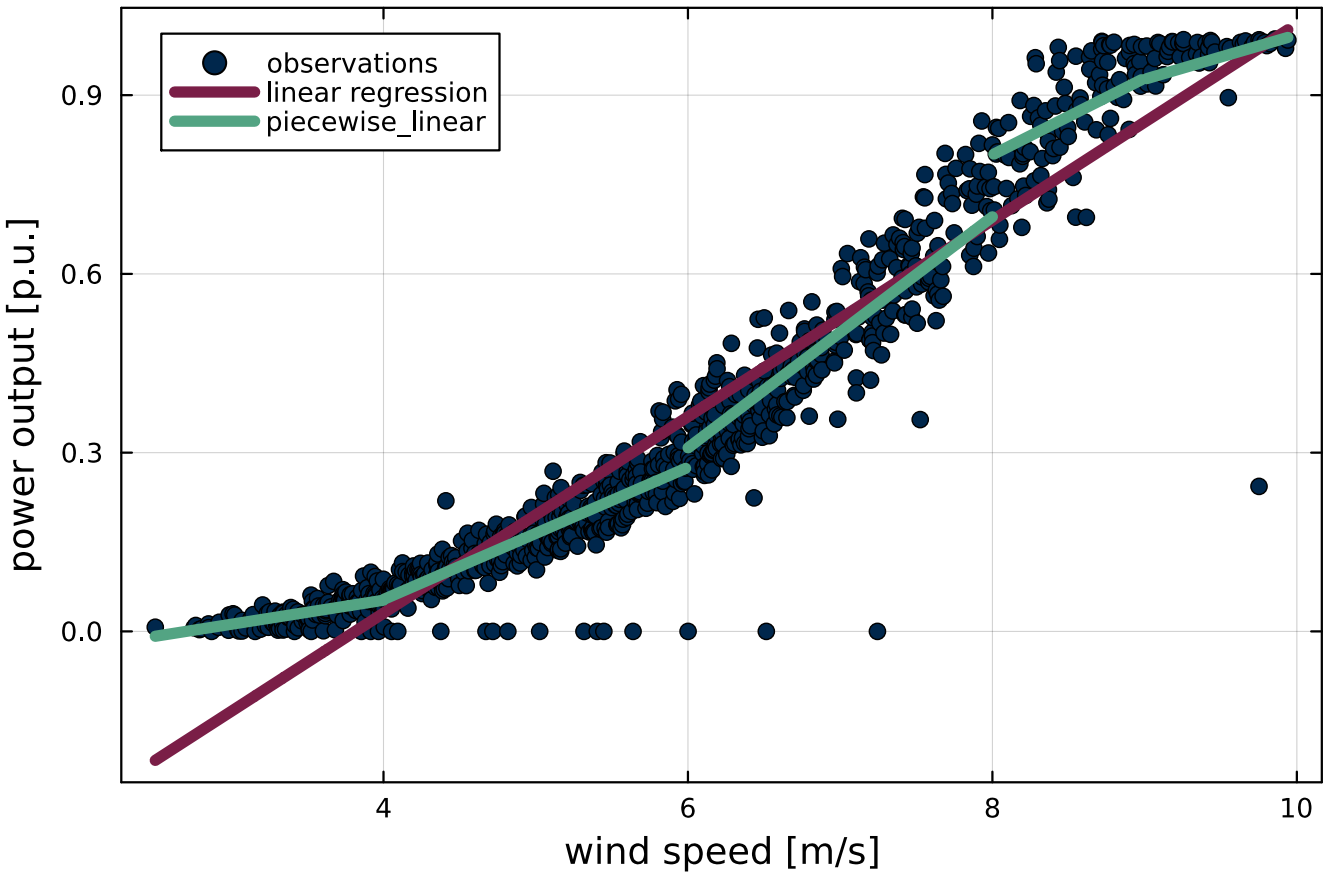
- For $\alpha = 0.01$, `iter_max`$= 25,000$ and intial model parameters $(w, b) = (0, 0)$:



- Can we make it better?

Linear regression results

- For $\alpha = 0.01$, `iter_max`$= 25,000$ and intial model parameters $(w, b) = (0, 0)$:



- Can we make it better? **Yes! Let's fit several regression models, each for a particular wind speed range**

Quiz time

Which statement about wind power forecasting is FALSE?

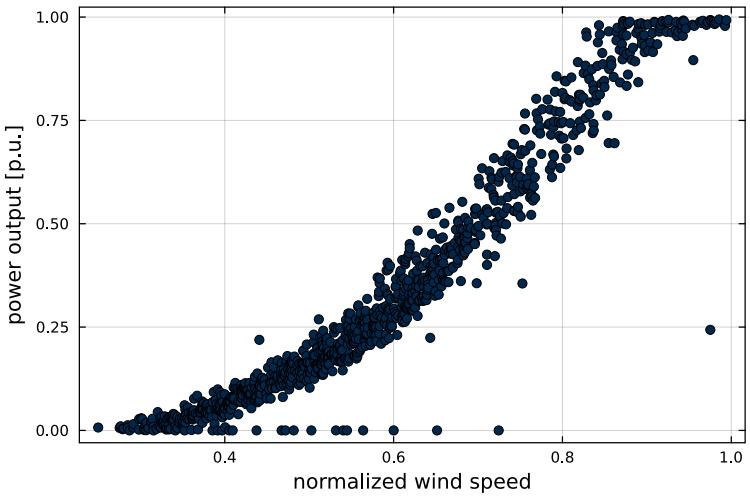
A Trying to predict renewable power output, we will also be wrong

B The point forecast can be used by ISOs for reserve determination

C The linear regression model captures the ascending trend of the wind power curve

D The interval forecast is less informative than predictive density

Feature transformation



Wind power extraction

$$p(u) = \frac{1}{2} C_p \rho A u^3$$

A rotor swept area
u wind speed

A rotor swept area
u wind speed

- We used linear regression despite the wind power curve is non-linear
- Leverage the underlying physical law to transform the features:
  - Add cubic wind speed[1]
  - More generally, fit polynomial or logistic function[2]
  - In any case, the model remains linear in features
- Is there a more flexible (less restrictive) approach to curve fitting?

[1] Chapter 3 of Burton, T., Jenkins, N., Sharpe, D., & Bossanyi, E. Wind energy handbook. 2011

[2] Wang, Yun, et al. Wind power curve modeling and wind power forecasting with inconsistent data. 2018

Fitting using Radial basis functions

- Radial basis function (**RBF**) is real-valued function φ whose value depends only on the (typically Euclidean) distance between the input and some fixed point
- For a fixed point $\mathbf{c}$ (center), the RBF is $\varphi_{\mathbf{c}}(\mathbf{x}) = \hat{\varphi}(\|\mathbf{x} - \mathbf{c}\|)$
- A series $\varphi_{\mathbf{c}_1}(\mathbf{x}), \ldots, \varphi_{\mathbf{c}_k}(\mathbf{x})$ forms the basis for some function space of interest
- The weighted sum of RBFs approximates the function $f(\mathbf{x})$ of interest, i.e.,

$$f(\mathbf{x}) = \sum_{i=1}^{k} w_i \varphi_{\mathbf{c}_i}(\mathbf{x}) = \sum_{i=1}^{k} w_i \hat{\varphi}(\|\mathbf{x} - \mathbf{c}_i\|)$$

where w_i is the weight to be optimized (e.g., solving a linear regression)

Fitting using Radial basis functions

- Radial basis function (**RBF**) is real-valued function φ whose value depends only on the (typically Euclidean) distance between the input and some fixed point
- For a fixed point $\mathbf{c}$ (center), the RBF is $\varphi_{\mathbf{c}}(\mathbf{x}) = \hat{\varphi}(\|\mathbf{x} - \mathbf{c}\|)$
- A series $\varphi_{\mathbf{c}_1}(\mathbf{x}), \ldots, \varphi_{\mathbf{c}_k}(\mathbf{x})$ forms the basis for some function space of interest
- The weighted sum of RBFs approximates the function $f(\mathbf{x})$ of interest, i.e.,

$$f(\mathbf{x}) = \sum_{i=1}^{k} w_i \varphi_{\mathbf{c}_i}(\mathbf{x}) = \sum_{i=1}^{k} w_i \hat{\varphi}(\|\mathbf{x} - \mathbf{c}_i\|)$$

  where w_i is the weight to be optimized (e.g., solving a linear regression)
- Examples of relevant RBFs (kernels):
  - Gaussian

$$\hat{\varphi}(r) = \exp\left(-(\gamma r)^2\right)$$

  - Thin plate spline

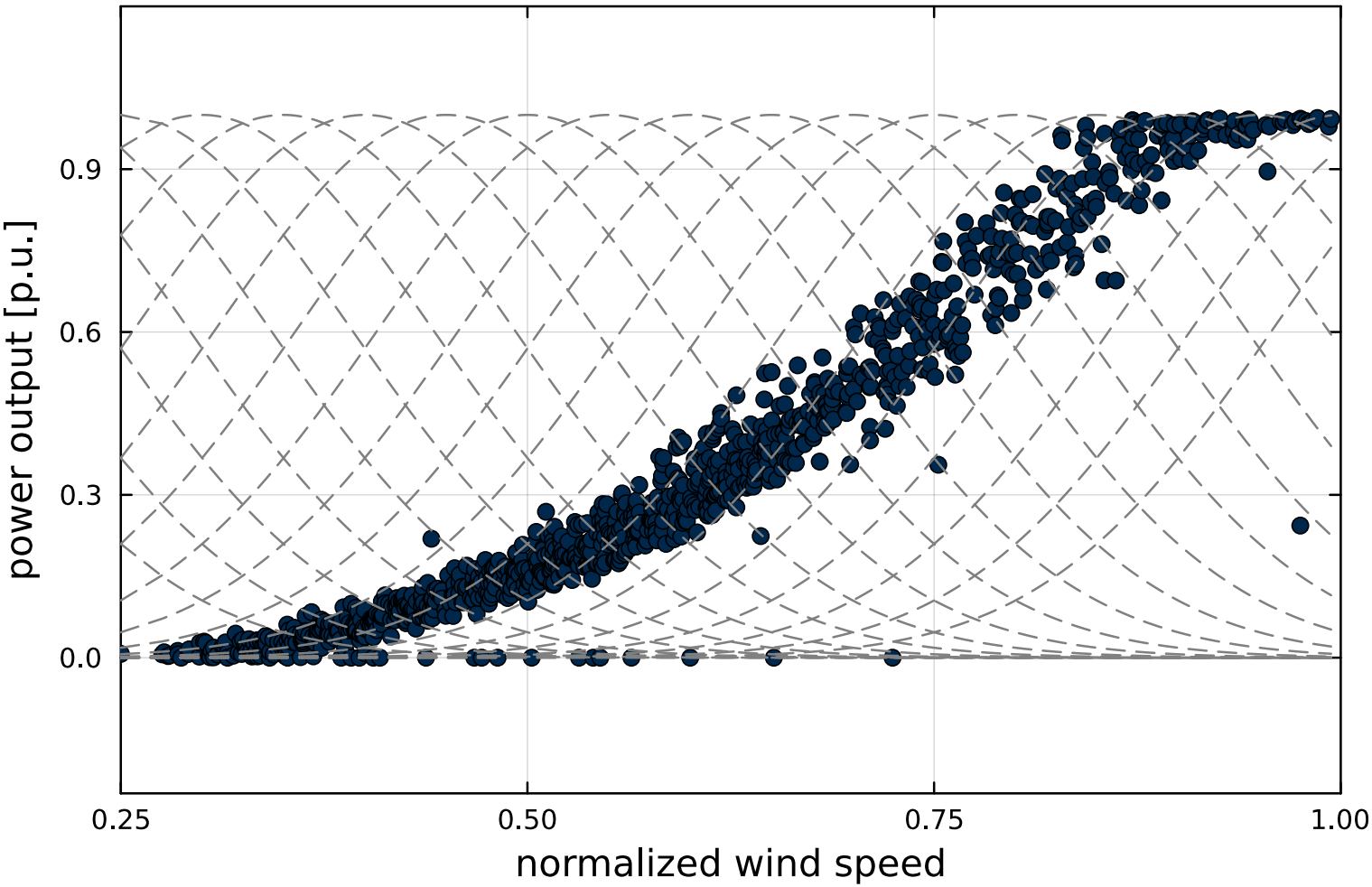
$$\hat{\varphi}(r) = r^2 \ln(r)$$

  - Inverse quadratic

$$\frac{1}{1 + (\gamma r)^2}$$

  - Inverse multiquadric

$$\frac{1}{\sqrt{1 + (\gamma r)^2}}$$



The weight of each Gaussian RBF is 1. Our goal is to weight them properly to shape a smooth wind power curve.

RBF-Based Support Vector Regression

- Dataset $\{(x_i, t_i)\}_{i=1}^n$, with feature x and target t, we estimate the model[3]

$$y = w_0 + \sum_{j=1}^{k} w_j \varphi_j(x)$$

with parameters $\mathbf{w} = [w_0, w_1, \ldots, w_k]^\top$

- Optimize the model by solving the following convex optimization

$$\underset{\mathbf{w}}{\text{minimize}} \quad \underbrace{\frac{1}{2n} \sum_{i=1}^{n} \left(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i \right)^2}_{\text{cost function } \mathcal{C}}$$

Q. Is it convex or non-convex optimization problem?

[3] Z. Jianwu, W. Qiao. Support vector machine-based short-term wind power forecasting. 2011.

RBF-Based Support Vector Regression

- Dataset $\{(x_i, t_i)\}_{i=1}^n$, with feature x and target t, we estimate the model[3]

$$y = w_0 + \sum_{j=1}^{k} w_j \varphi_j(x)$$

with parameters $\mathbf{w} = [w_0, w_1, \ldots, w_k]^\top$

- Optimize the model by solving the following convex optimization

$$\underset{\mathbf{w}}{\text{minimize}} \quad \underbrace{\frac{1}{2n} \sum_{i=1}^{n} \left(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i \right)^2}_{\text{cost function } \mathcal{C}}$$

Q. Is it convex or non-convex optimization problem?

- Optimization is not convex in features $\mathbf{x}$, but convex in RBF weights $\mathbf{w}$!

[3] Z. Jianwu, W. Qiao. Support vector machine-based short-term wind power forecasting. 2011.

Regularization

- To prevent overfitting, we introduce a regularization term $\mathcal{R}(\mathbf{w})$

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2n} \sum_{i=1}^{n} \left(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i \right)^2 + \lambda \mathcal{R}(\mathbf{w})$$

with typically small parameter λ, $e.g.$, 10^{-5}

Regularization

- To prevent overfitting, we introduce a regularization term $\mathcal{R}(\mathbf{w})$

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2n} \sum_{i=1}^{n} \left(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i \right)^2 + \lambda \mathcal{R}(\mathbf{w})$$

  with typically small parameter λ, *e.g.*, 10^{-5}

- **Tikhonov regularization**: $\mathcal{R}(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|_2^2$ gives rise to ridge regression

- For the fitting models with many RBFs, Tikhonov regularization mitigates the problem of multicollinearity (highly correlated RBFs with similar parameters)

Regularization

- To prevent overfitting, we introduce a regularization term $\mathcal{R}(\mathbf{w})$

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2n} \sum_{i=1}^{n} \left(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i \right)^2 + \lambda \mathcal{R}(\mathbf{w})$$

  with typically small parameter $\lambda, e.g., 10^{-5}$

- **Tikhonov regularization**: $\mathcal{R}(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|_2^2$ gives rise to ridge regression
- For the fitting models with many RBFs, Tikhonov regularization mitigates the problem of multicollinearity (highly correlated RBFs with similar parameters)
- ℓ_1**-regularization**: $\mathcal{R}(\mathbf{w}) = \|\mathbf{w}\|_1$ gives rise to LASSO regression
- LASSO: Least Absolute Shrinkage and Selection Operator
- For the fitting models with many RBFs, LASSO puts zero weight on redundant features (introduce many RBFs and select only important ones using LASSO)

Solving RBF-Based Support Vector Regression

- Closed-form solution is unlikely due to dimensionality of the feature matrix $\mathbf{X}$

$$\mathbf{X} = \begin{bmatrix} \mathbf{1} & \varphi_1(\mathbf{x}) & \dots & \varphi_k(\mathbf{x}) \end{bmatrix}$$

- The gradient descent is also challenging due to dimensionality of the gradients

$$w_0 \leftarrow w_0 - \frac{\alpha}{n} \sum_{i=1}^{n} \left(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i \right)$$

$$w_j \leftarrow w_j - \frac{\alpha}{n} \sum_{i=1}^{n} \left(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i \right) \varphi_j(x_i) \quad \forall j = 1, \dots, k$$

which is very expensive to compute at **every** iterations

Q. How can we avoid computing so many gradients at every iterations?

Stochastic gradient descent (SGD)

for $k = 1, \ldots,$ `iter_max` **do**

step 1. Sample index $i \sim \mathcal{U}[1, n]$ from a uniform distribution

step 2. Update model parameters on (x_i, t_i) only

$$w_0 \leftarrow w_0 - \alpha\Big(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i\Big)$$

$$w_j \leftarrow w_j - \alpha\Big(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i\Big)\varphi_j(x_i) \quad \forall j = 1, \ldots, k$$

end for

- At each iteration, the gradients are only evaluated at a single x_i
- Regarded as a stochastic approximation of the original gradient descent
- Mini-batch extension: sample a small subset of training data

Stochastic gradient descent (SGD)

for $k = 1, \ldots,$ `iter_max` **do**

step 1. Sample index $i \sim \mathcal{U}[1, n]$ from a uniform distribution

step 2. Update model parameters on (x_i, t_i) only

$$w_0 \leftarrow w_0 - \alpha\Big(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i\Big)$$

$$w_j \leftarrow w_j - \alpha\Big(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i\Big)\varphi_j(x_i) \quad \forall j = 1, \ldots, k$$

end for

- At each iteration, the gradients are only evaluated at a single x_i
- Regarded as a stochastic approximation of the original gradient descent
- Mini-batch extension: sample a small subset of training data

Q. Does it account for regularization?

Tikhonov regularization for SGD

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\left(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i\right)^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

Q. How do the gradients in SGD look like? (in-class exercise)

Tikhonov regularization for SGD

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\Big(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i\Big)^2 + \frac{\lambda}{2}\,\|\mathbf{w}\|_2^2$$

Q. How do the gradients in SGD look like? (in-class exercise)

$$w_0 \leftarrow w_0 - \alpha\Big(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i - \lambda w_0\Big)$$

$$w_j \leftarrow w_j - \alpha\Big(\Big(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i\Big)\varphi_j(x_i) - \lambda w_j\Big) \quad \forall j = 1, \ldots, k$$

Lasso regularization for SGD

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\left(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i\right)^2 + \lambda \left\|\mathbf{w}\right\|_1$$

Lasso regularization for SGD

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\left(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i\right)^2 + \lambda \|\mathbf{w}\|_1$$

- Focus on the terms involving w_0

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\Bigg(\underbrace{w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i}_{\text{residual } r_i}\Bigg)^2 + \lambda |w_0|$$

Lasso regularization for SGD

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\left(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i\right)^2 + \lambda \|\mathbf{w}\|_1$$

- Focus on the terms involving w_0

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\Bigg(\underbrace{w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i}_{\text{residual } r_i}\Bigg)^2 + \lambda |w_0|$$

- Derivative of the squared term w.r.t. w_0

$$\frac{\partial}{\partial w_0} \frac{1}{2} r_i^2 = r_i$$

Lasso regularization for SGD

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\left(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i\right)^2 + \lambda \|\mathbf{w}\|_1$$

- Focus on the terms involving w_0

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\underbrace{\left(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i\right)}_{\text{residual } r_i}^2 + \lambda |w_0|$$

- Derivative of the squared term w.r.t. w_0

$$\frac{\partial}{\partial w_0} \frac{1}{2} r_i^2 = r_i$$

- For the regularization term, the **subgradient** is

$$\frac{\partial}{\partial w_0} \lambda |w_0| = \begin{cases} \lambda, & w_0 > 0 \\ -\lambda, & w_0 < 0 \\ \text{any } g \in [-\lambda, \lambda], & w_0 = 0 \end{cases}$$

Lasso regularization for SGD

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\Big(w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i\Big)^2 + \lambda \|\mathbf{w}\|_1$$

- Focus on the terms involving w_0

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\Big(\underbrace{w_0 + \sum_{j=1}^{k} w_j \varphi_j(x_i) - t_i}_{\text{residual } r_i}\Big)^2 + \lambda |w_0|$$

- Derivative of the squared term w.r.t. w_0

$$\frac{\partial}{\partial w_0} \frac{1}{2} r_i^2 = r_i$$

- For the regularization term, the **subgradient** is

$$\frac{\partial}{\partial w_0} \lambda |w_0| = \begin{cases} \lambda, & w_0 > 0 \\ -\lambda, & w_0 < 0 \\ \text{any } g \in [-\lambda, \lambda], & w_0 = 0 \end{cases}$$

- The optimality conditions w.r.t. w_0

$$r_i + \lambda \text{sign}(w_0) = 0, \quad w_0 \neq 0 \quad \text{and} \quad |r_i| \leqslant \lambda, \quad w_0 = 0$$

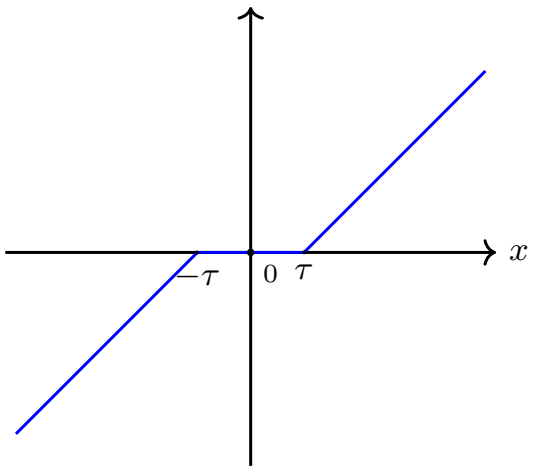
Lasso regularization for SGD (cont'd)

- We need a well defined optimality condition for the gradient update, i.e.,

$$w_0 \leftarrow w_0 - \alpha r_i \qquad \text{[w/o } \ell_1\text{-regularization]}$$

- Instead, we will apply a **soft-thresholding** function[4]

$$S_\tau(x) = \begin{cases} x - \tau, & x > \tau \\ 0, & |x| \leqslant \tau \\ x + \tau, & x < -\tau \end{cases}$$



- Applying the soft-thresholding function to the gradient step

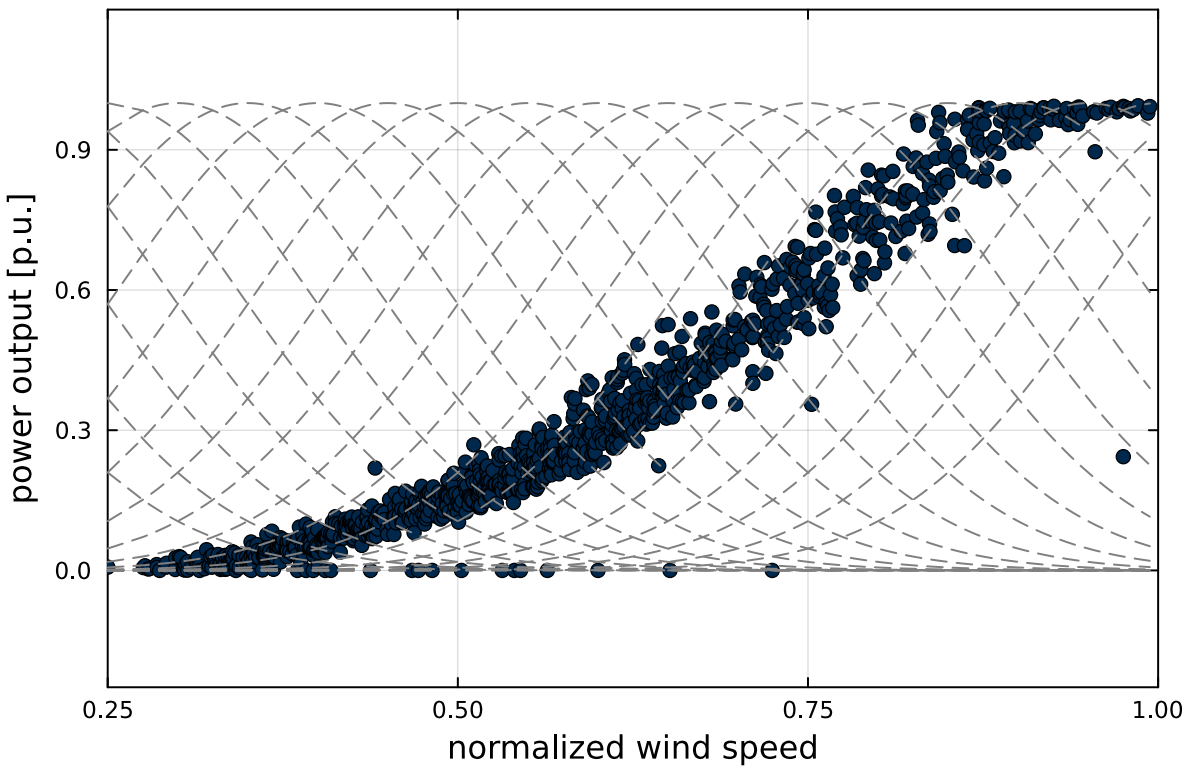
$$w_0 = S_{\alpha\lambda}(w_0 - \alpha r_i) = \begin{cases} w_0 - \alpha r_i - \alpha\lambda, & w_0 - \alpha r_i > \alpha\lambda \\ 0, & |w_0 - \alpha r_i| \leqslant \alpha\lambda \\ w_0 - \alpha r_i + \alpha\lambda, & w_0 - \alpha r_i < -\alpha\lambda \end{cases}$$

- This ensures small w_0 shrink to 0, as desired. Apply similarly to $w_1, \ldots, w_k$

[4] Wright and Yi. High-dimensional data analysis with low-dimensional models. 2022.

RBF-SVM for wind power curve fitting

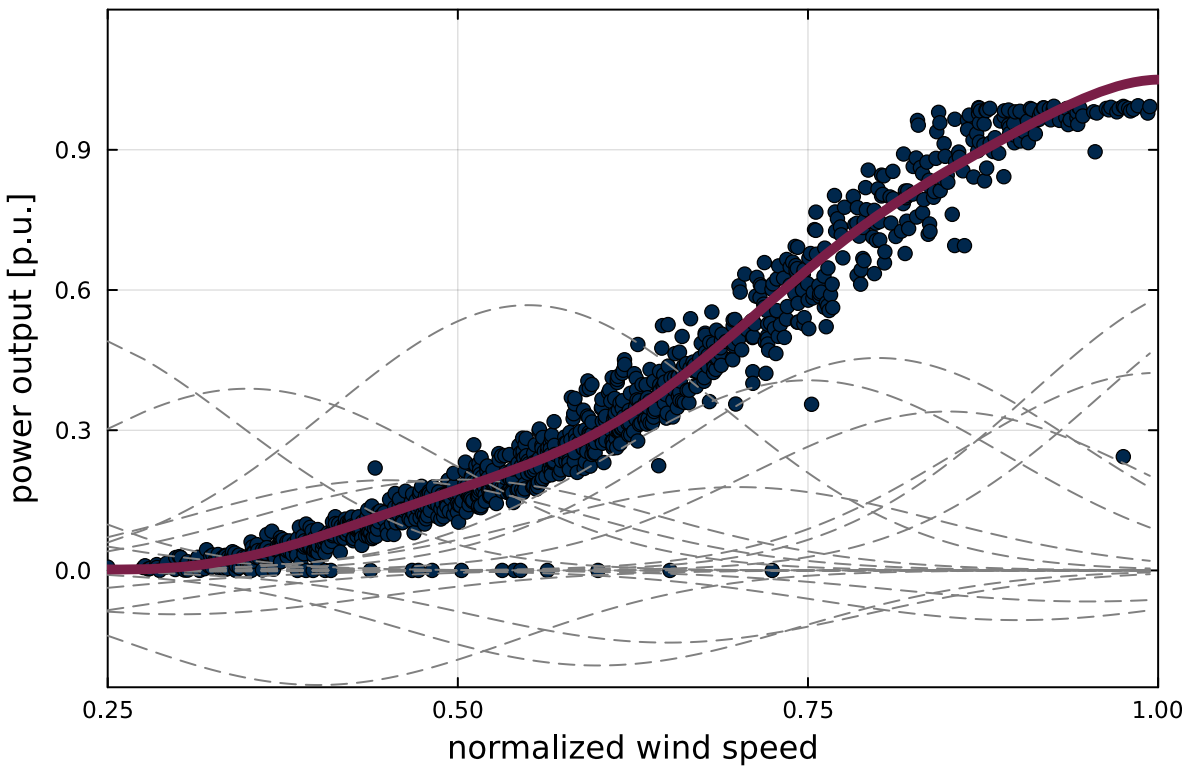
- Gaussian radial basis functions $\varphi_i(x) = \exp\left(-(5\,\|x - c_i\|)^2\right)$
- Centered across the normalized wind speed range $c = 0 : 0.05 : 1.1$
- step size $\alpha = 0.01$, `iter_max` $= 25,000$, reg. parameter $\lambda = 10^{-3}$
- Gray lines depict the 23 Gaussian RBF for curve fitting



the weight of each RBF is 1

RBF-SVM for wind power curve fitting

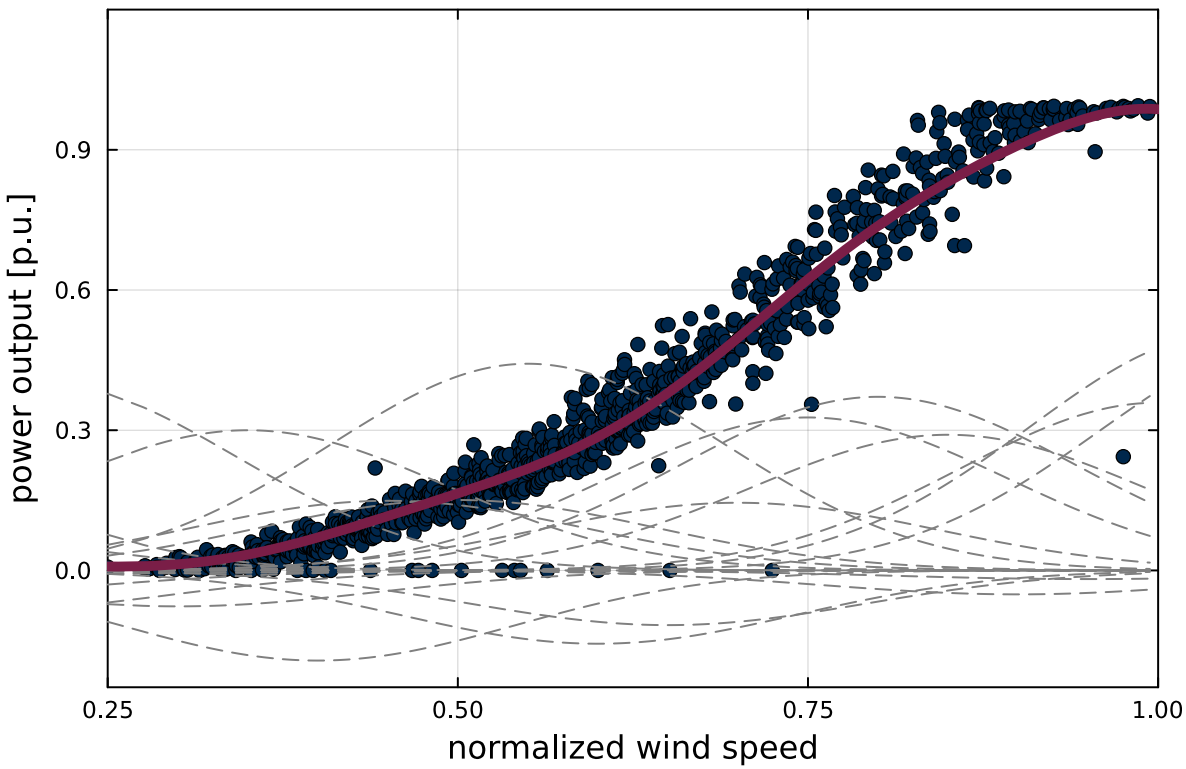
- Gaussian radial basis functions $\varphi_i(x) = \exp\left(-(5\,\|x - c_i\|)^2\right)$
- Centered across the normalized wind speed range $c = 0 : 0.05 : 1.1$
- step size $\alpha = 0.01$, `iter_max` $= 25,000$, reg. parameter $\lambda = 10^{-3}$
- Gray lines depict the 23 Gaussian RBF for curve fitting



no regularization

RBF-SVM for wind power curve fitting

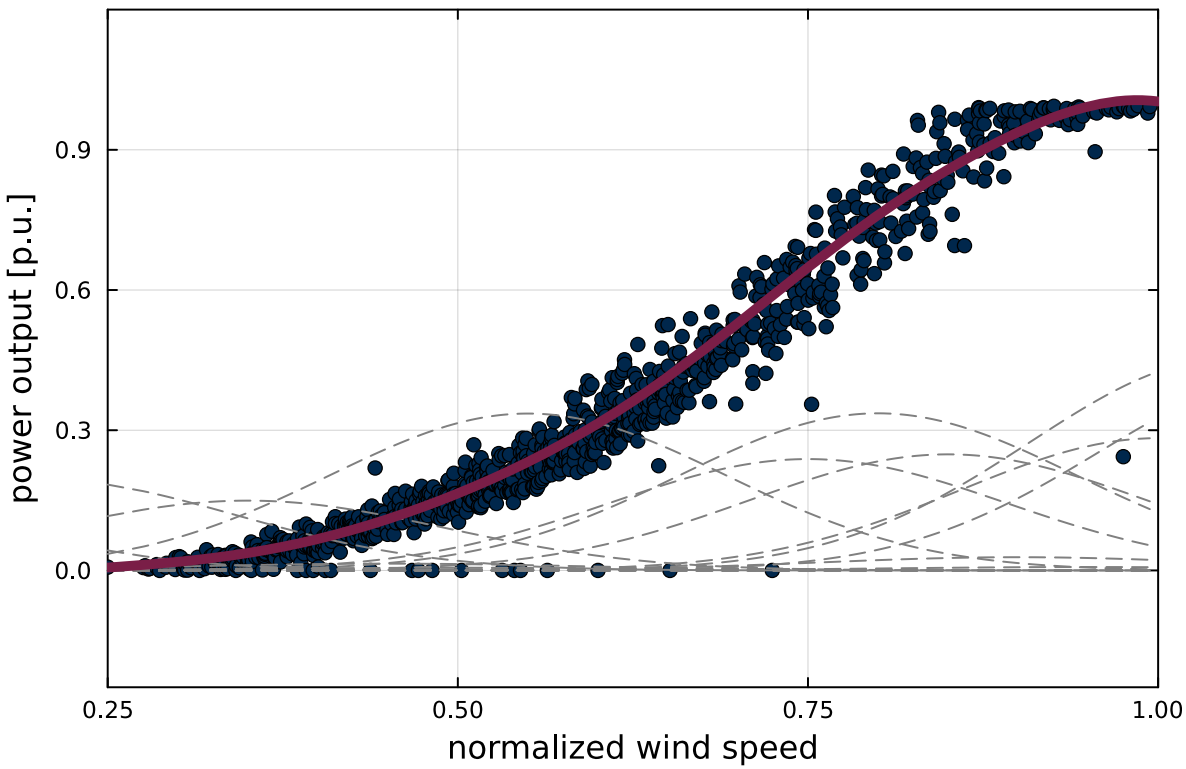
- Gaussian radial basis functions $\varphi_i(x) = \exp\left(-(5\,\|x - c_i\|)^2\right)$
- Centered across the normalized wind speed range $c = 0 : 0.05 : 1.1$
- step size $\alpha = 0.01$, `iter_max` $= 25,000$, reg. parameter $\lambda = 10^{-3}$
- Gray lines depict the 23 Gaussian RBF for curve fitting



Tikhonov regularization

RBF-SVM for wind power curve fitting

- Gaussian radial basis functions $\varphi_i(x) = \exp\left(-(5\,\|x - c_i\|)^2\right)$
- Centered across the normalized wind speed range $c = 0 : 0.05 : 1.1$
- step size $\alpha = 0.01$, `iter_max` $= 25,000$, reg. parameter $\lambda = 10^{-3}$
- Gray lines depict the 23 Gaussian RBF for curve fitting



Lasso regularization

Resources

- Wang, Y., Hu, Q., Srinivasan, D., & Wang, Z. (2018). Wind power curve modeling and wind power forecasting with inconsistent data. IEEE Transactions on Sustainable Energy, 10(1), 16-25.

- Pinson, P., Chevallier, C., & Kariniotakis, G. N. (2007). Trading wind generation from short-term probabilistic forecasts of wind power. IEEE transactions on Power Systems, 22(3), 1148-1156.

- Wright, John, and Yi Ma. High-dimensional data analysis with low-dimensional models: Principles, computation, and applications. Cambridge University Press, 2022. **[page 316]**

- Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.