Regression (overview)

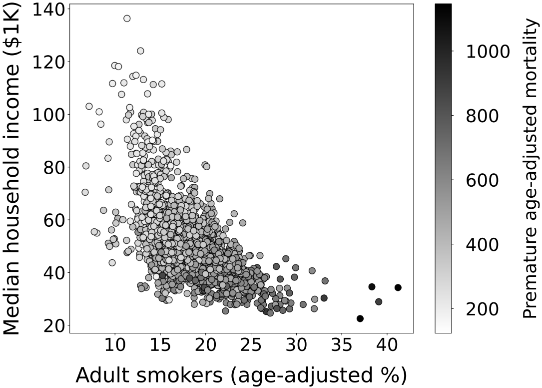
Probability and Statistics for Data Science

Carlos Fernandez-Granda

These slides are based on the book Probability and Statistics for Data Science by Carlos Fernandez-Granda, available for purchase here. A free preprint, videos, code, slides and solutions to exercises are available at https://www.ps4ds.net

Regression

Goal: Estimate response from features



Response: Premature mortality

Features:

(1) Fraction of adult smokers (2) Median household income

Probabilistic formulation

Find function h, such that $h(x)$ approximates the response $\tilde{y}$ when the features $\tilde{x} = x$

How do we evaluate the estimator?

Mean squared error (MSE)

$$\text{MSE}(h) := \mathrm{E}\left[(\tilde{y} - h(\tilde{x}))^2\right]$$

Optimal estimator: Conditional mean

$$\mu_{\tilde{y}\,|\,\tilde{x}} = \arg\min_{h} \text{MSE}(h)$$

In practice

Data: (x_1, y_1), (x_2, y_2), ..., (x_n, y_n)

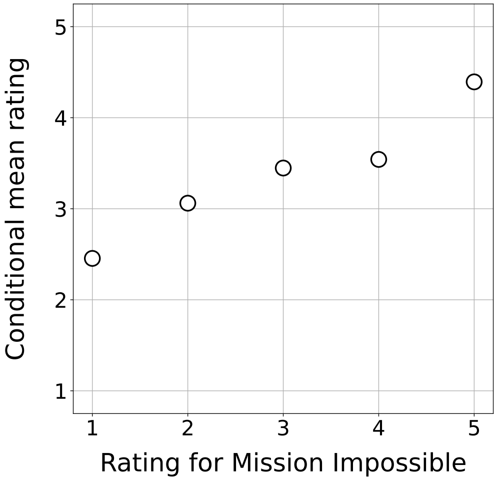
Residual sum of squares (RSS)

$$\text{RSS}(h) := \sum_{i=1}^{n} (y_i - h(x_i))^2$$

Optimal estimator: Empirical conditional mean

Movie rating

Estimate rating for Independence Day given rating for Mission Impossible



Are we done here? No!

Approximating conditional mean is often impossible due to curse of dimensionality

To predict from 100 movie ratings, how many different feature vectors?

$5^{100} > 10^{68}$!

Feature vector is likely to be unique

We need to make assumptions

Plan

Linear Regression

Overfitting and Regularization

Nonlinear Regression

Linear Regression

Overfitting and Regularization

Nonlinear Regression

Linear regression

We approximate the response as an affine function of the features

$$\tilde{y} \approx \ell(\tilde{x}) := \sum_{i=1}^{d} \beta[i]\tilde{x}[i] + \alpha$$
$$= \beta^T \tilde{x} + \alpha$$

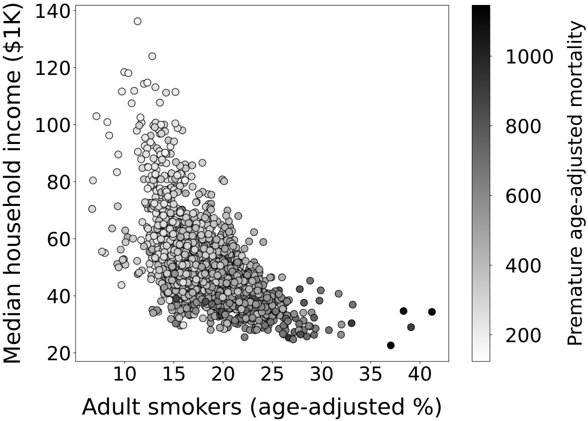
Linear minimum MSE (MMSE) estimator

Linear minimum MSE estimator of response $\tilde{y}$ given features $\tilde{x}$

$$\ell_{\text{MMSE}}(\tilde{x}) = \Sigma_{\tilde{x}\tilde{y}}^{T} \Sigma_{\tilde{x}}^{-1} \left(\tilde{x} - \mu_{\tilde{x}}\right) + \mu_{\tilde{y}}$$

Optimal if features and response are jointly Gaussian

Data



Response: Premature mortality

Features:

(1) Fraction of adult smokers (2) Median household income

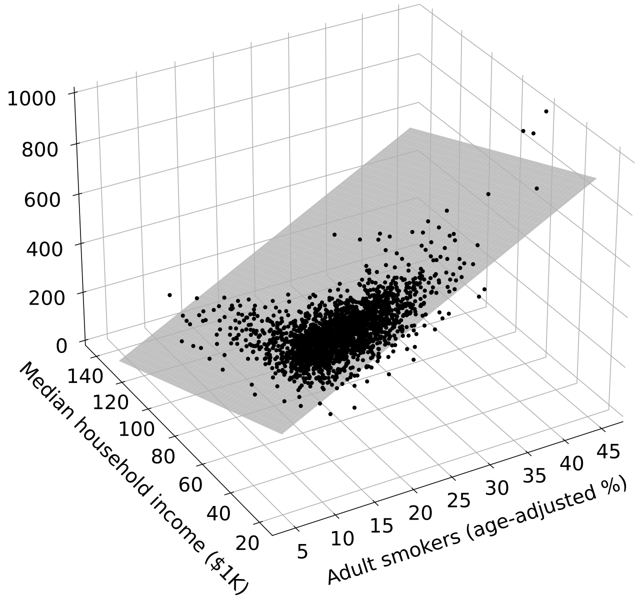
Ordinary least squares

Dataset: (x_1, y_1), (x_2, y_2), ..., (x_n, y_n)

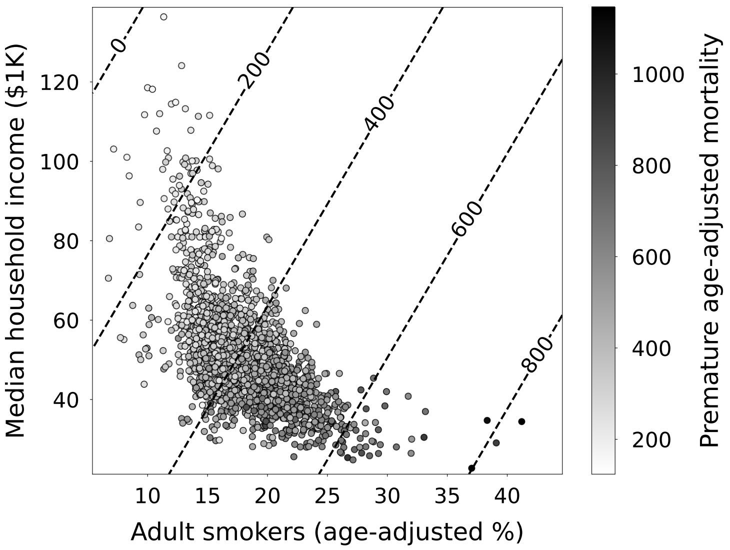
$$(\beta_{\text{OLS}}, \alpha_{\text{OLS}}) := \arg\min_{\beta,\alpha} \sum_{i=1}^{n} \left(y_i - \beta^T x_i - \alpha\right)^2$$

$$\begin{aligned}\ell_{\text{OLS}}(x_i) &= \beta_{\text{OLS}}^T x_i + \alpha_{\text{OLS}} \\ &= \Sigma_{XY}^T \Sigma_X^{-1} (x_i - m(X)) + m(Y)\end{aligned}$$

$$15.7\,x_{\text{tobacco}} - 3.04\,x_{\text{income}} + 281$$



$$15.7\,x_{\text{tobacco}} - 3.04\,x_{\text{income}} + 281$$



Linear Regression

Overfitting and Regularization

Nonlinear Regression

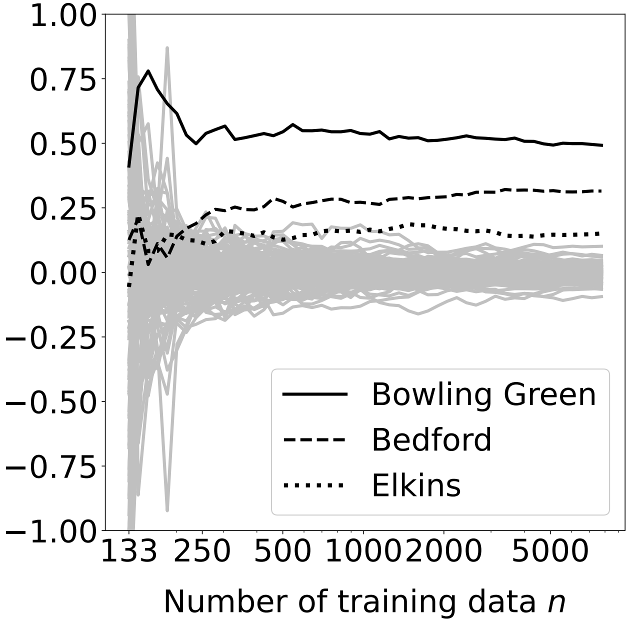
Temperature prediction

Response: Temperature in Versailles (Kentucky)

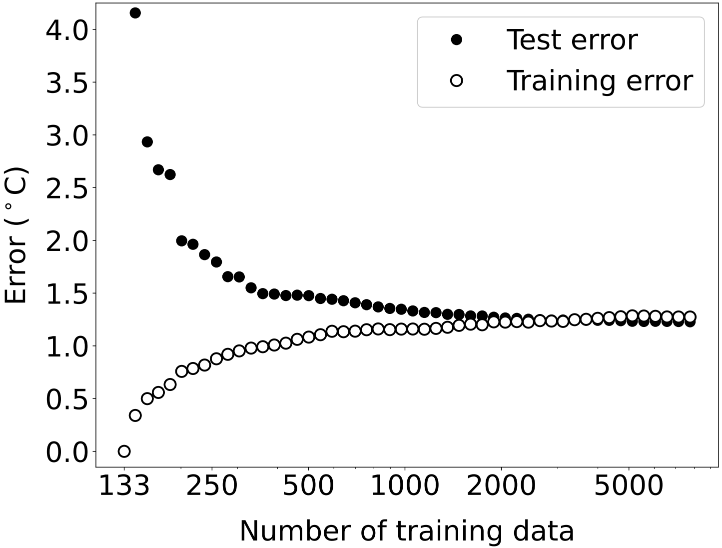
Features: Temperatures at 133 other locations

OLS coefficients (large n)

OLS coefficients



Training and test error



Linear response with random additive noise

$$\tilde{y}_{\text{train}} := X_{\text{train}} \beta_{\text{true}} + \tilde{z}$$

$$X_{\text{train}} := \begin{bmatrix} x_1^T \\ x_2^T \\ \cdots \\ x_n^T \end{bmatrix}$$

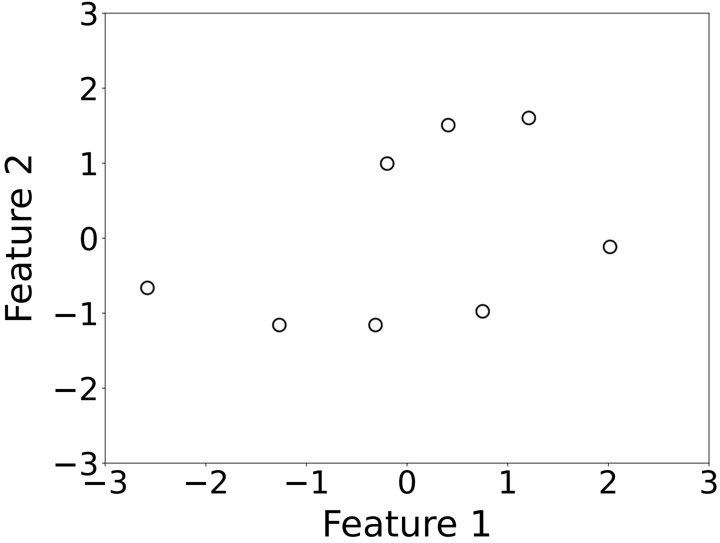
Noise $\tilde{z}$ is i.i.d. with variance σ^2

Everything is centered to have zero mean

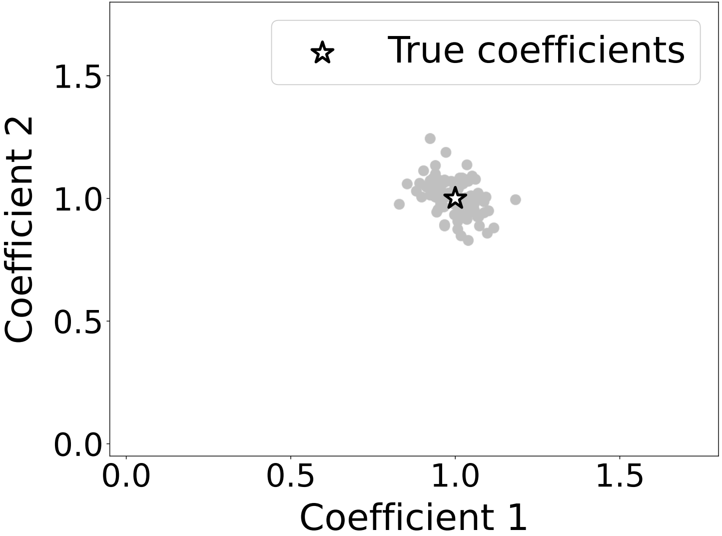
Properties of OLS coefficients

- Unbiased: Centered at true coefficients
- Consistent: Variance decreases as training data grows
- When features are collinear, variance is large in directions of low feature variance

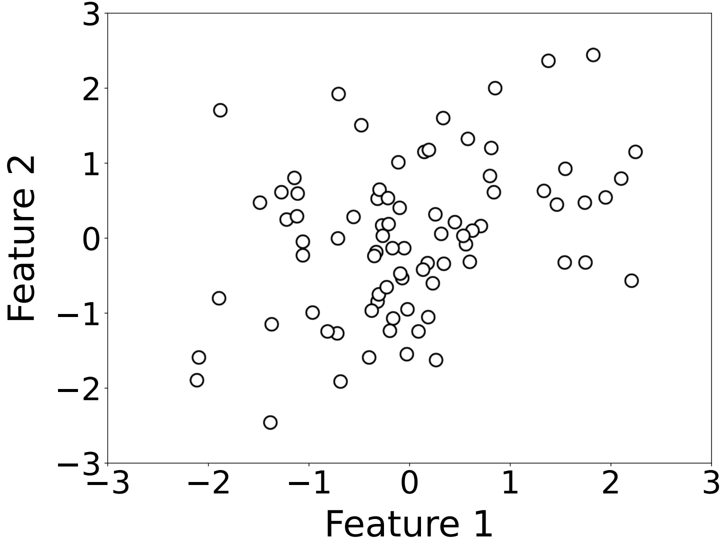
Features ($n := 8$)



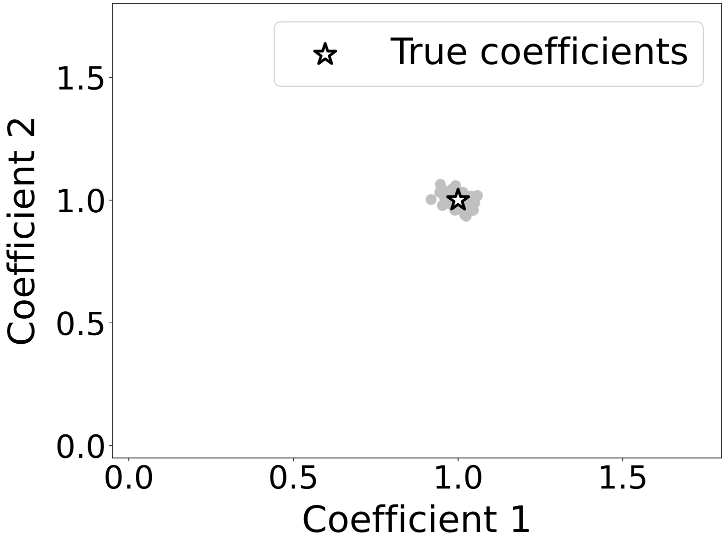
100 coefficient estimates



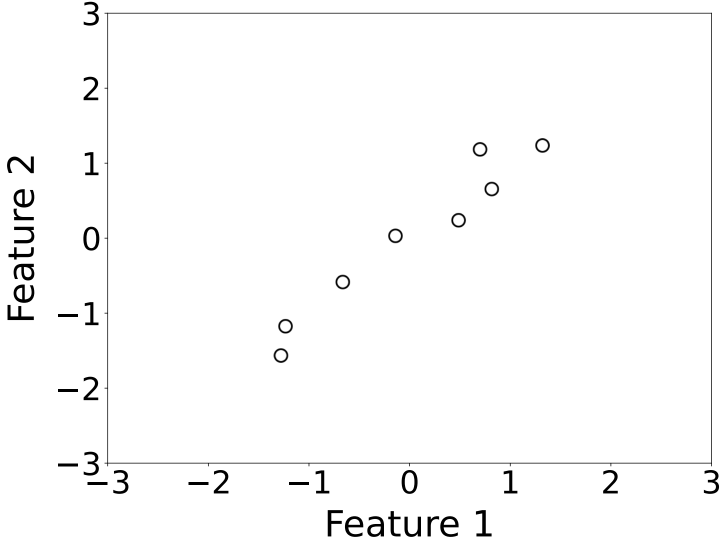
Features ($n := 80$)



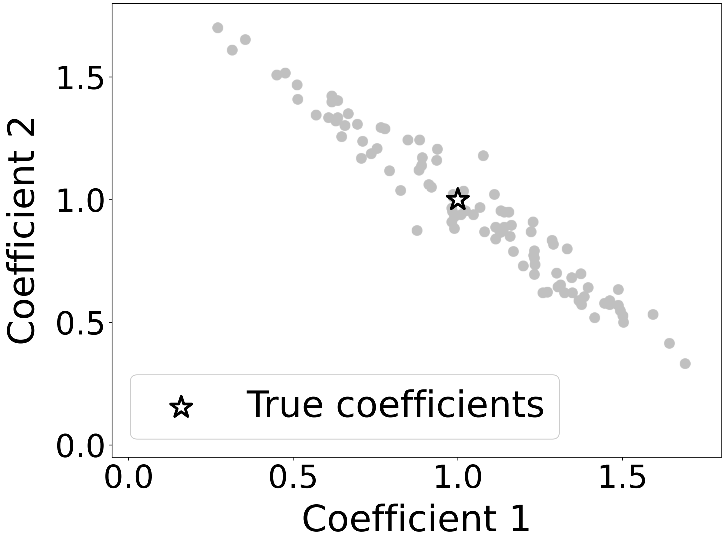
100 coefficient estimates



Collinear features ($n := 8$)



100 coefficient estimates



Ridge regression

Problem: For small n / collinear features, large OLS coefficients overfit the training data

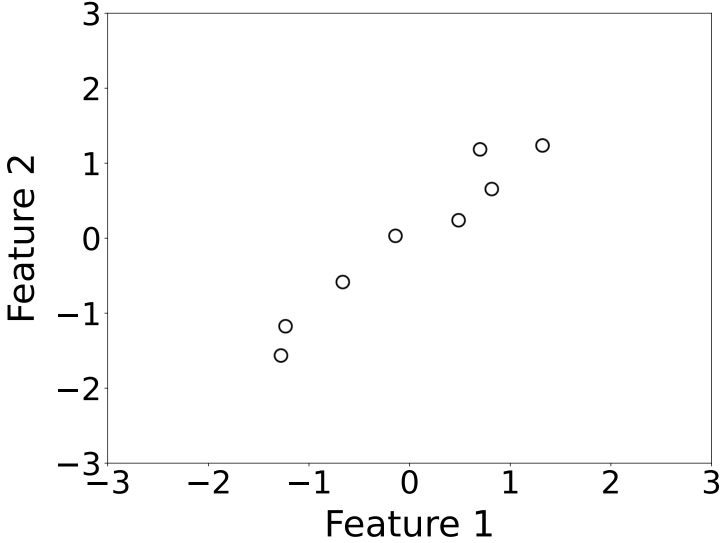
$$\beta_{\text{OLS}} = \arg\min_{\beta} \sum_{i=1}^{n} \left(y_i - \beta^T x_i \right)^2$$

Solution: Regularization, penalize the norm of the coefficients

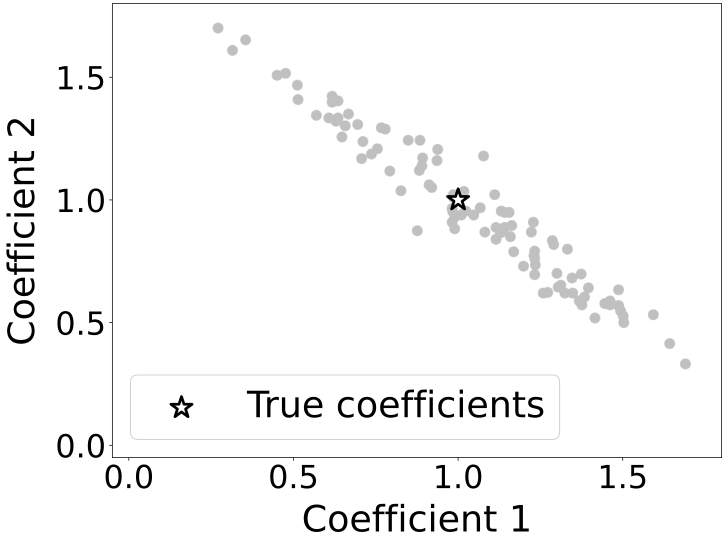
$$\beta_{\text{RR}} := \arg\min_{\beta} \sum_{i=1}^{n} \left(y_i - \beta^T x_i \right)^2 + \lambda \sum_{j=1}^{d} \beta_j^2$$

$\lambda > 0$ is a regularization parameter

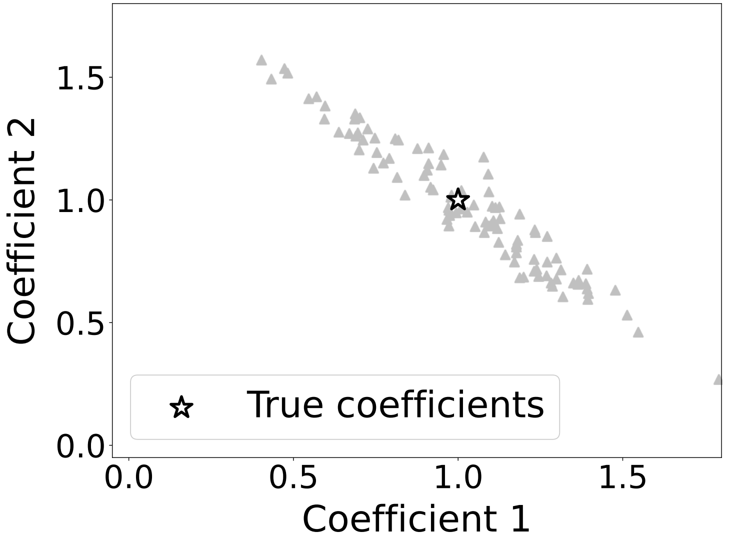
Collinear features ($n := 8$)



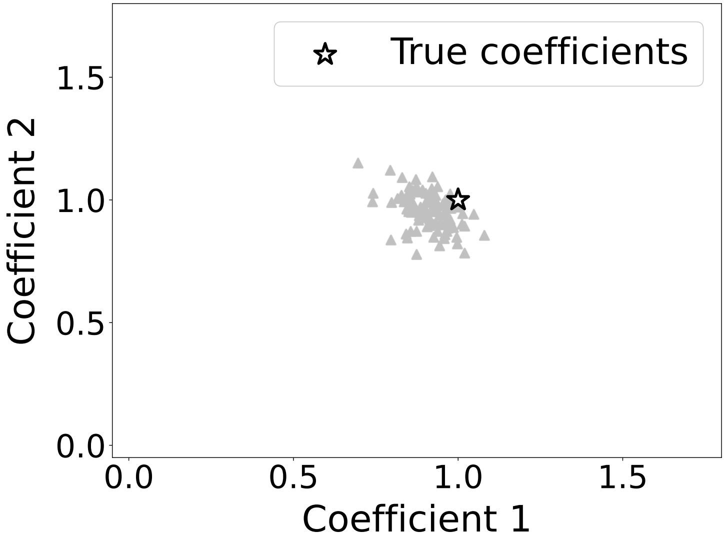
100 OLS coefficients



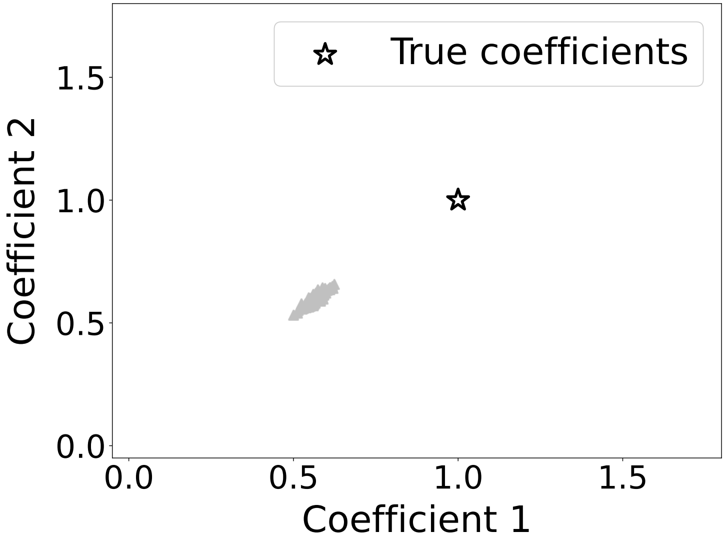
100 ridge-regression coefficients ($\lambda := 0.1$)



100 ridge-regression coefficients ($\lambda := 2$)



100 ridge-regression coefficients ($\lambda := 10$)

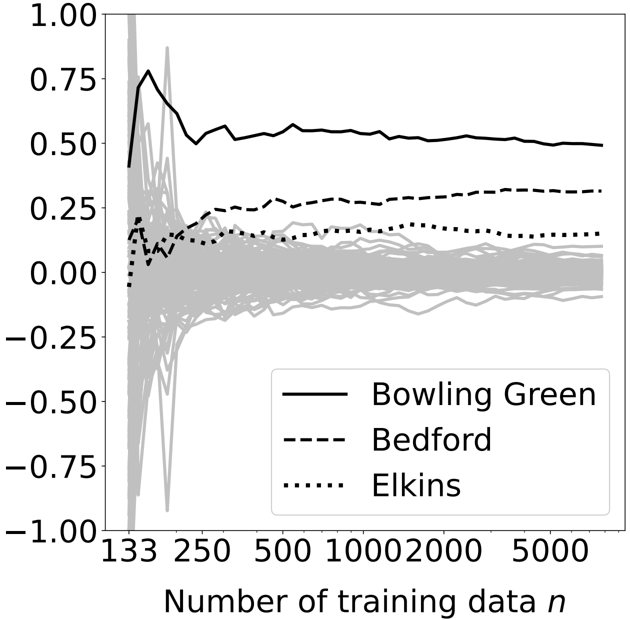


Properties of ridge regression

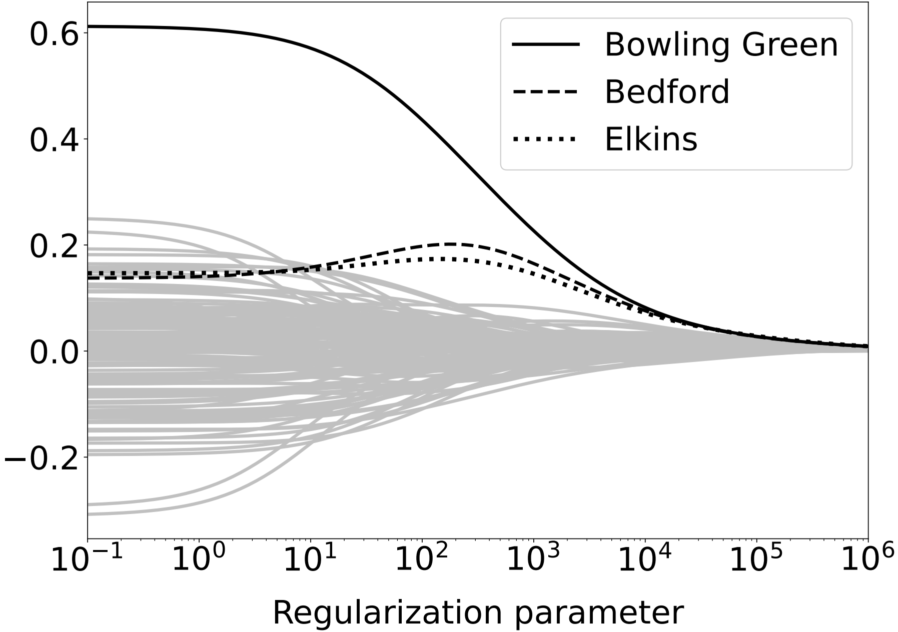
As λ increases,

- Variance decreases faster in directions of low feature variance, which prevents overfitting
- Bias towards zero

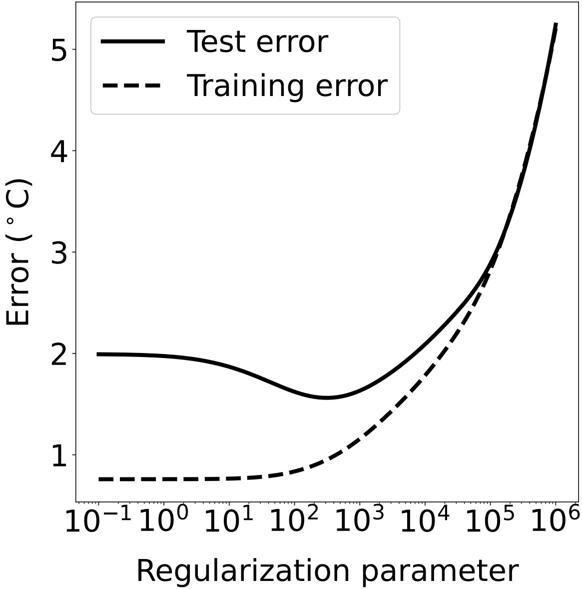
Temperature prediction: OLS coefficients



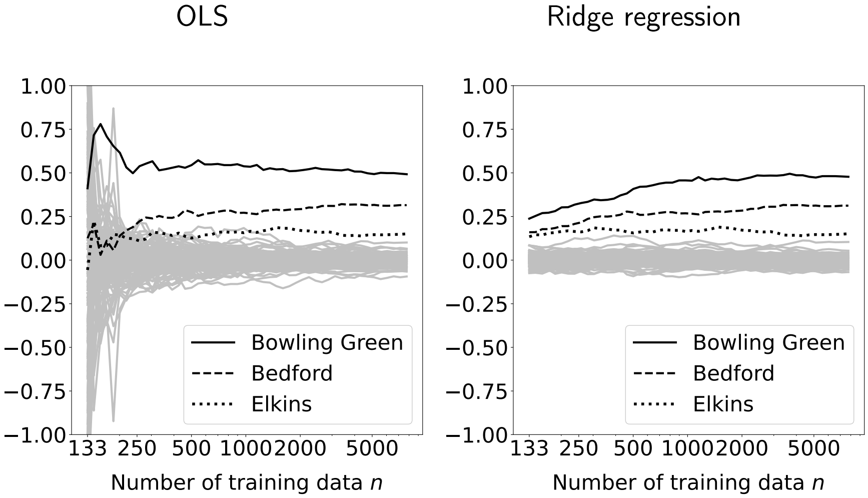
Ridge-regression coefficients ($n = 200$)



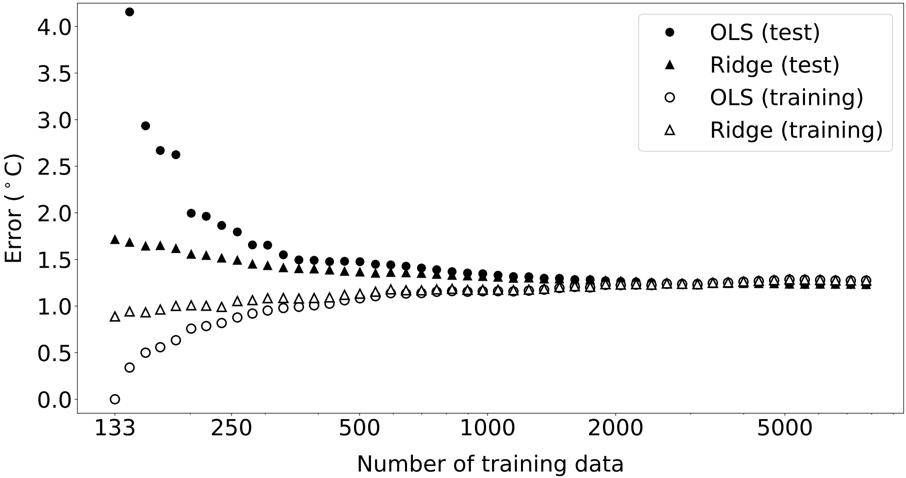
Training and test error ($n = 200$)



Coefficients



Error



OLS (test)
Ridge (test)
OLS (training)
Ridge (training)
Error (°C)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
133
250
500
1000
2000
5000
Number of training data

Sparse regression

Goal: Identify a small subset of features that provide a good fit

Equivalently, find sparse coefficients β that provide a good fit

Linear response with random additive noise

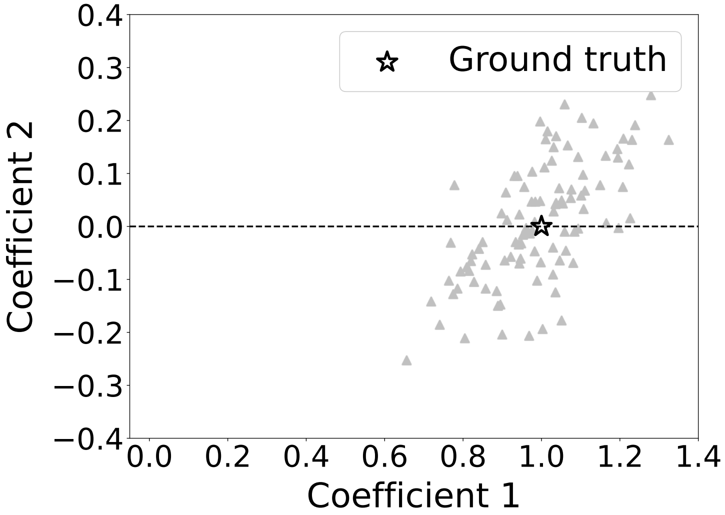
$$\tilde{y}_i := x_i[1] + \tilde{z} \quad 1 \leq i \leq n$$

$$X_{\text{train}} := \begin{bmatrix} x_1[1] & x_1[2] \\ x_2[1] & x_2[2] \\ \cdots & \\ x_n[1] & x_n[2] \end{bmatrix} \qquad \beta_{\text{true}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

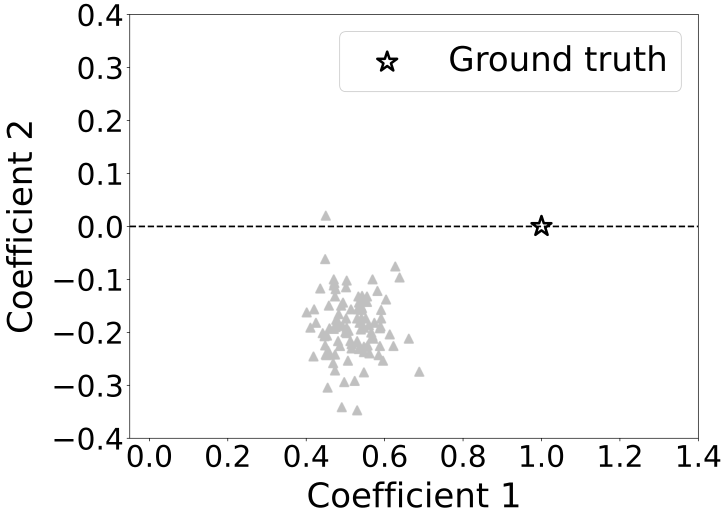
Noise $\tilde{z}$ is i.i.d. with fixed variance

Everything is centered to have zero mean

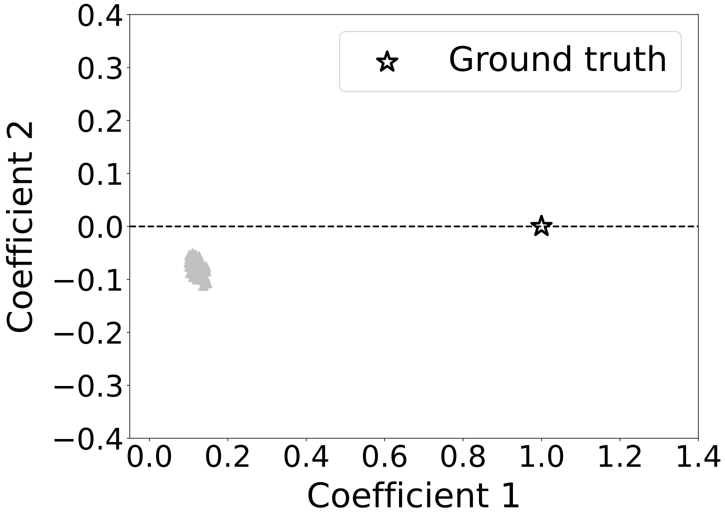
Ridge regression: Small λ



Ridge regression: Medium λ



Ridge regression: Large λ



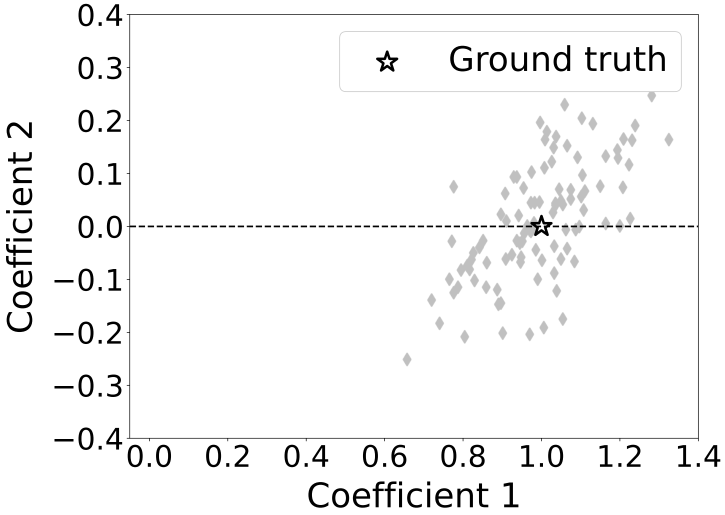
The lasso

Regularization penalizes the ℓ_1 norm of the coefficients (instead of ℓ_2 norm)

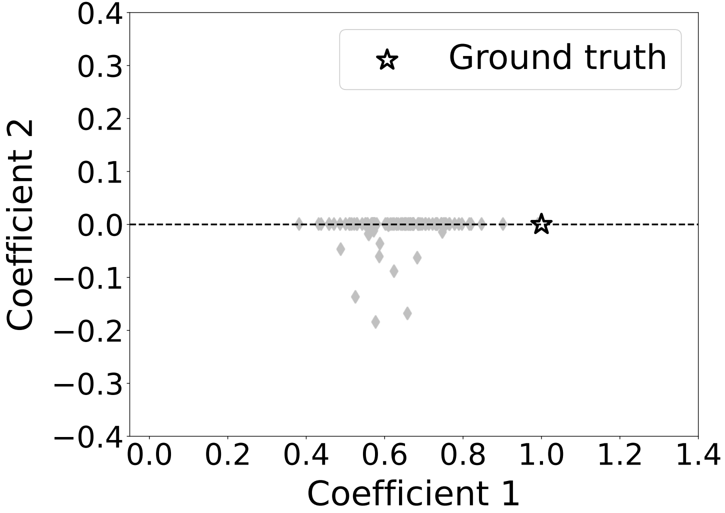
$$\beta_{\text{lasso}} := \arg\min_{\beta} \sum_{i=1}^{n} \left(y_i - \beta^T x_i \right)^2 + \lambda \, ||\beta||_1$$

$\lambda > 0$ is a regularization parameter

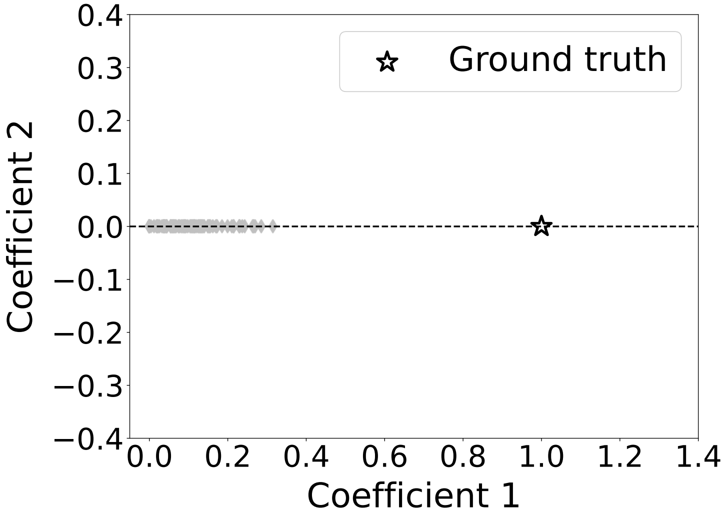
Lasso: Small λ



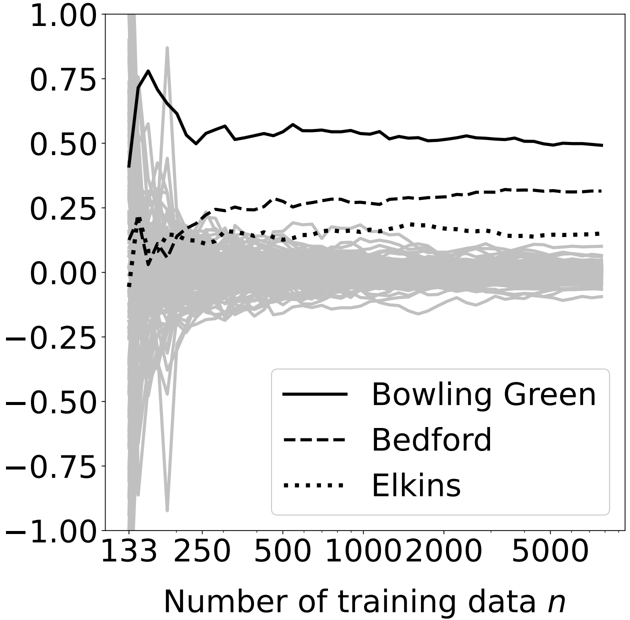
Lasso: Medium λ



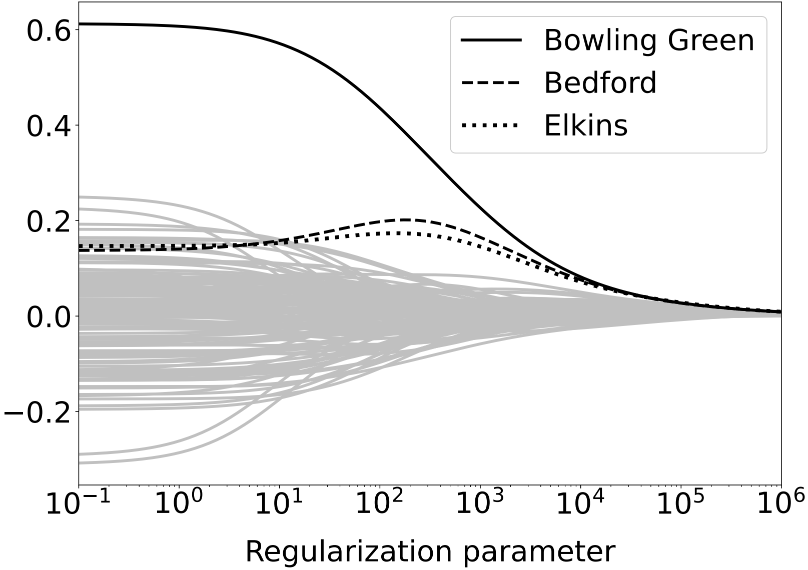
Lasso: Large λ



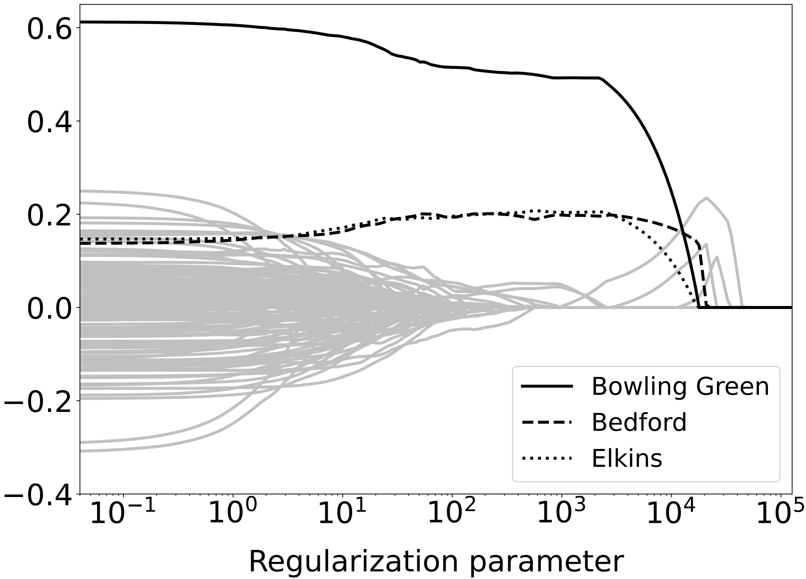
Temperature prediction: OLS coefficients



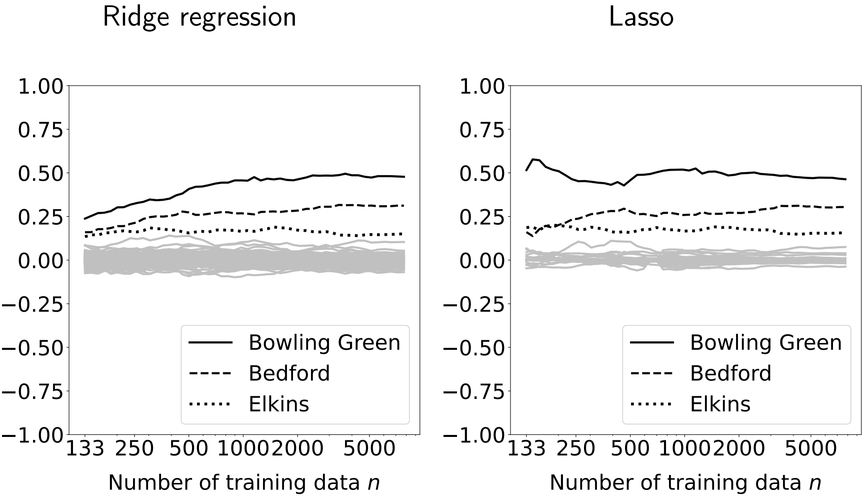
Ridge-regression coefficients ($n = 200$)



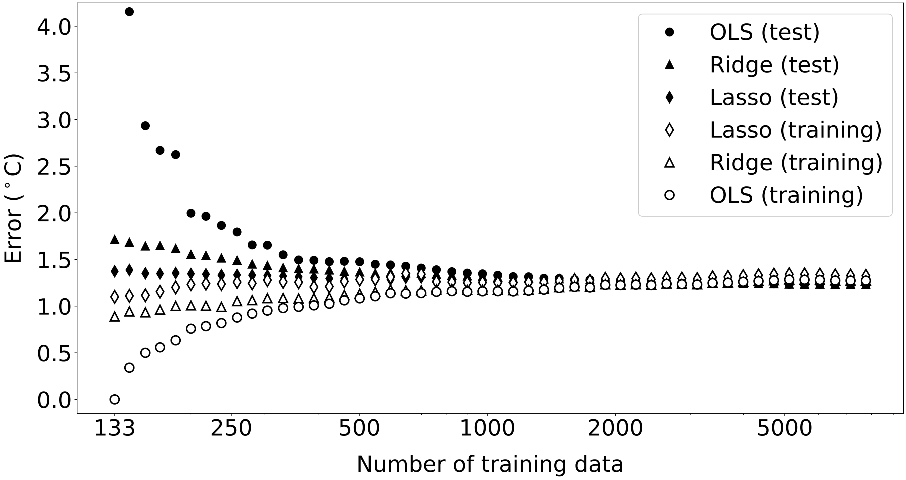
Lasso coefficients ($n = 200$)



Coefficients



Error



4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Error (°C)
133
250
500
1000
2000
5000
Number of training data
OLS (test)
Ridge (test)
Lasso (test)
Lasso (training)
Ridge (training)
OLS (training)

Linear Regression

Overfitting and Regularization

Nonlinear Regression

Example

Response: Temperature in Manhattan (Kansas)

Features:

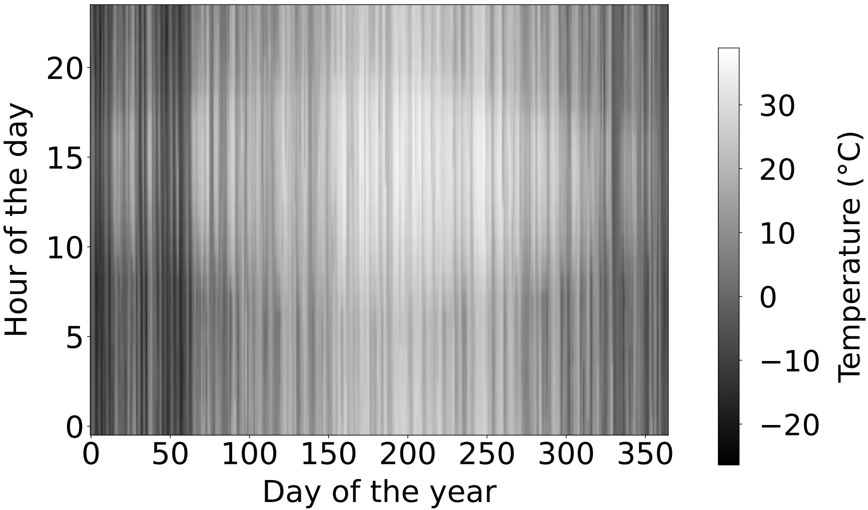
(1) Hour of the day (0-23)

(2) Day of the year (1-365)

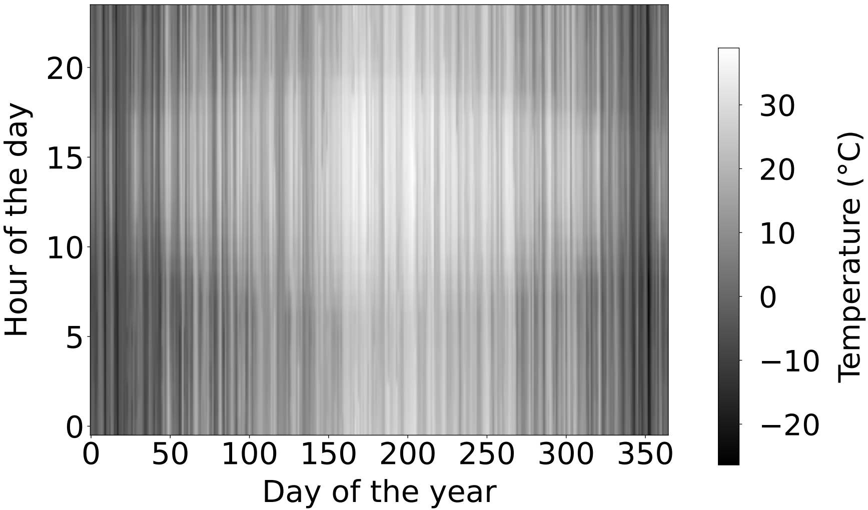
Training data: 2015

Test data: 2016

Training data

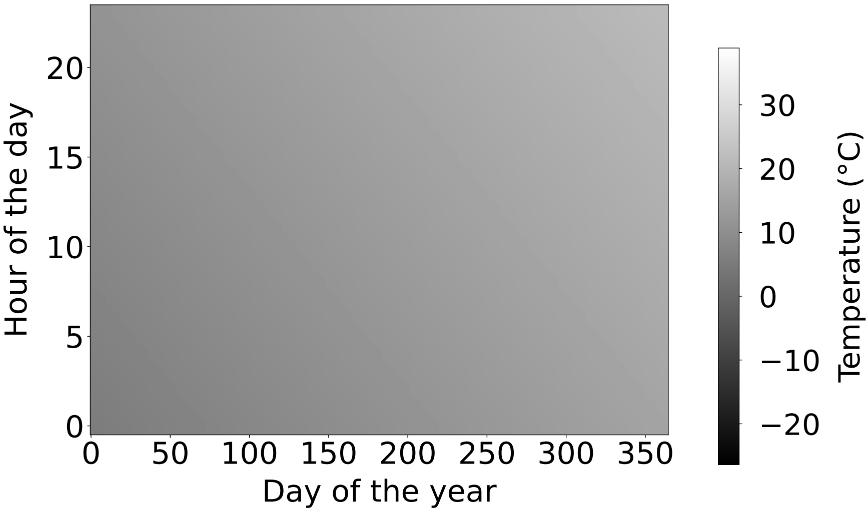


Test data



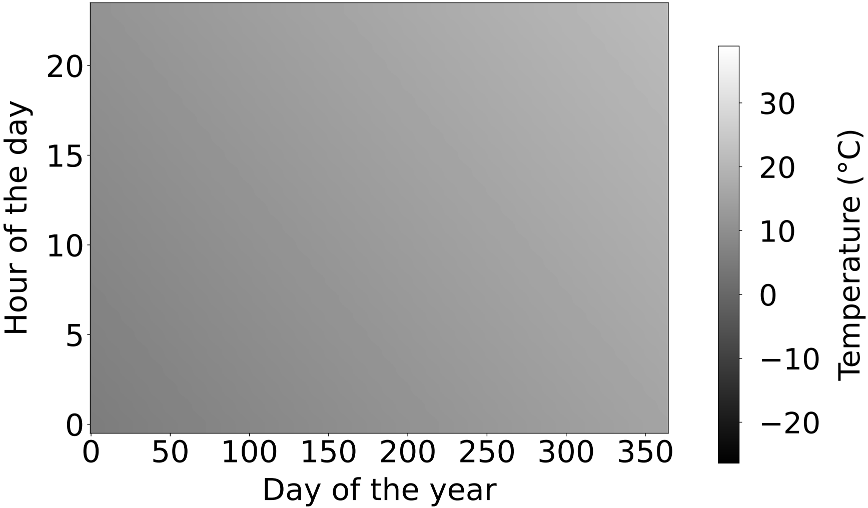
Hour of the day
0
5
10
15
20
Day of the year
0
50
100
150
200
250
300
350
Temperature (°C)
30
20
10
0
−10
−20

Linear model: 0.25 hour + 0.03 day + 5.85



Response increases or decreases proportionally to each feature (if we fix other features)

Linear model: 0.25 hour $+ 0.03$ day $+ 5.85$

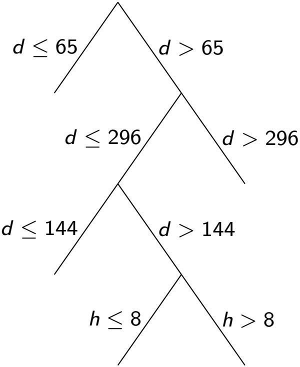


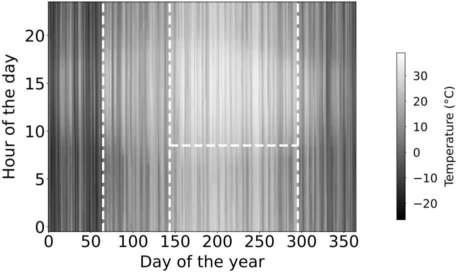
Training error: 10.8°C   Test error: 11.0°C

Nonlinear regression

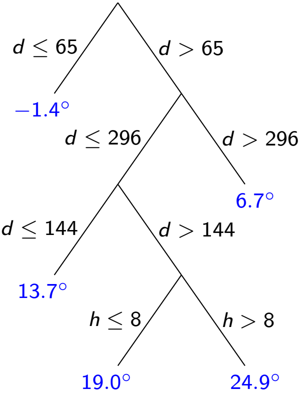
- Regression trees
- Tree ensembles
- Neural networks

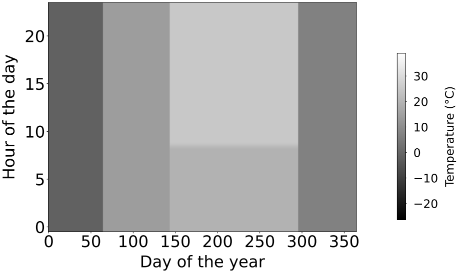
Regression tree



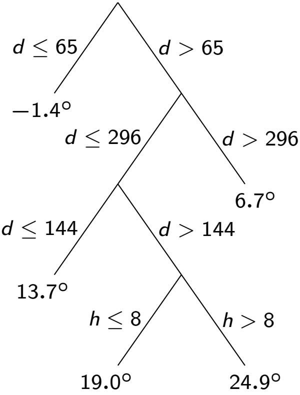


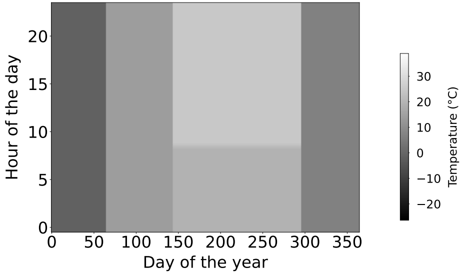
Regression tree



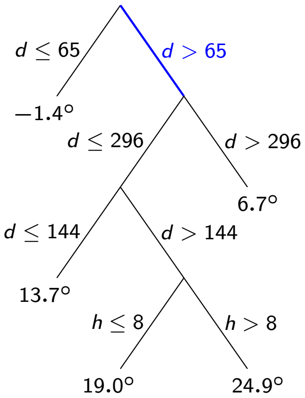


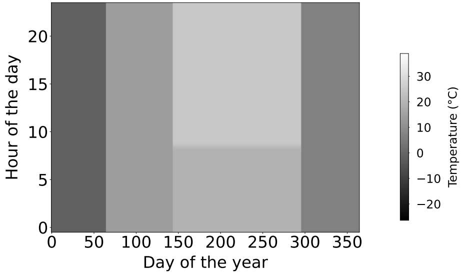
August 19 ($d := 251$) at 3 am ($h := 3$)?



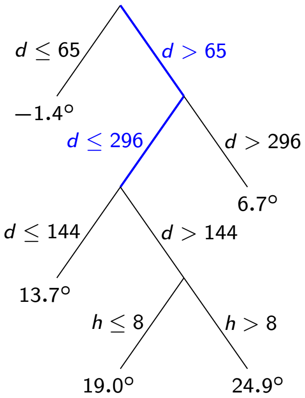


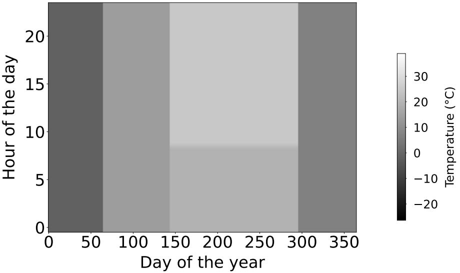
August 19 ($d := 251$) at 3 am ($h := 3$)?



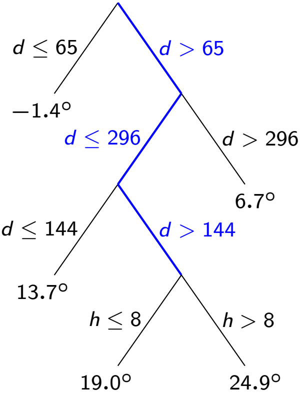


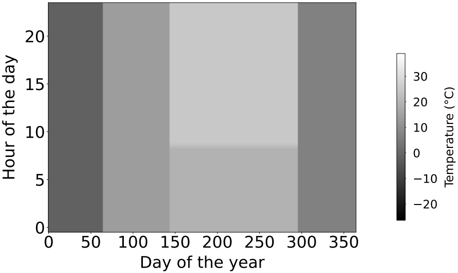
August 19 ($d := 251$) at 3 am ($h := 3$)?



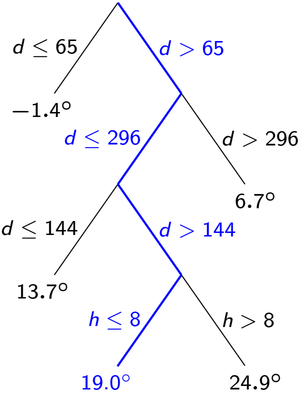


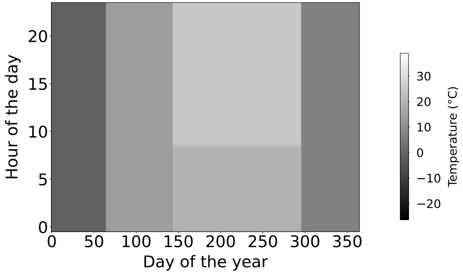
August 19 ($d := 251$) at 3 am ($h := 3$)?





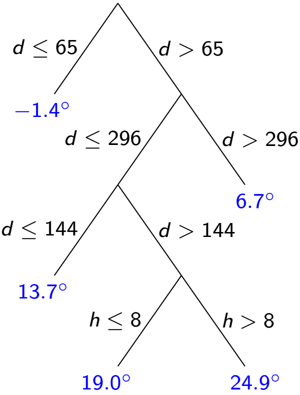
August 19 ($d := 251$) at 3 am ($h := 3$)?

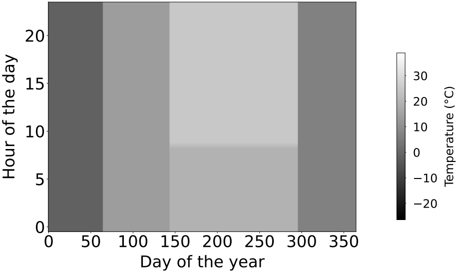




Interpretable!

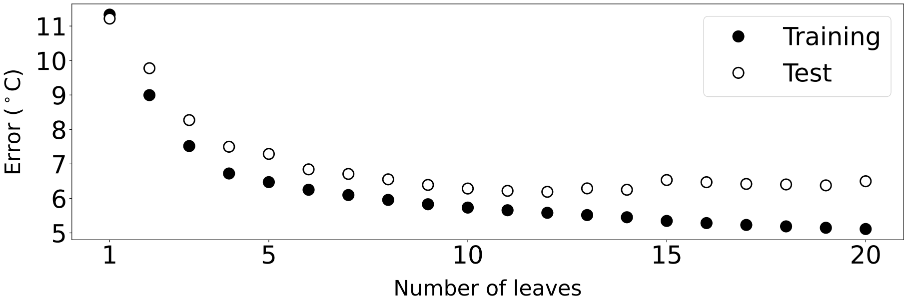
How do we build the tree?





Add bifurcations one by one to minimize training RSS

Training and test error



Ensembles

Problem: Simple trees underfit / Complex trees overfit

Solution: Combine multiple simple trees

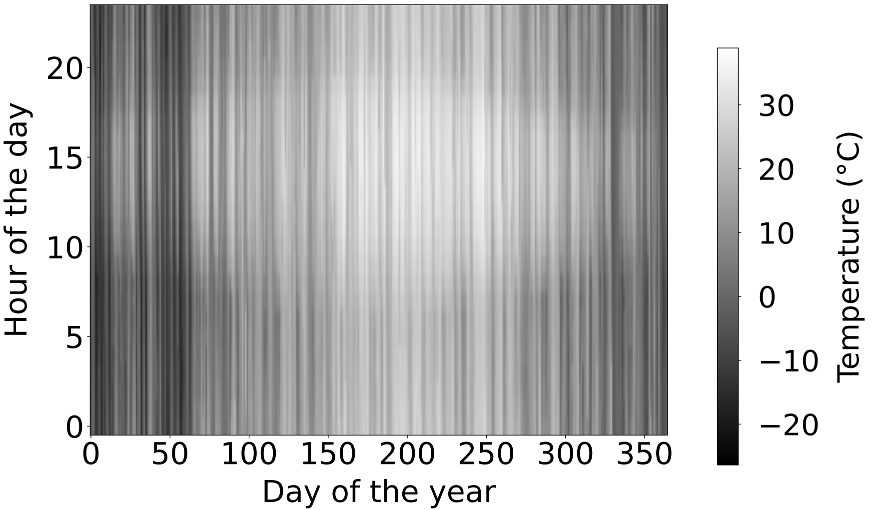
Three main strategies:

1. Bagging: Average trees trained on resampled datasets obtained via *bootstrapping*

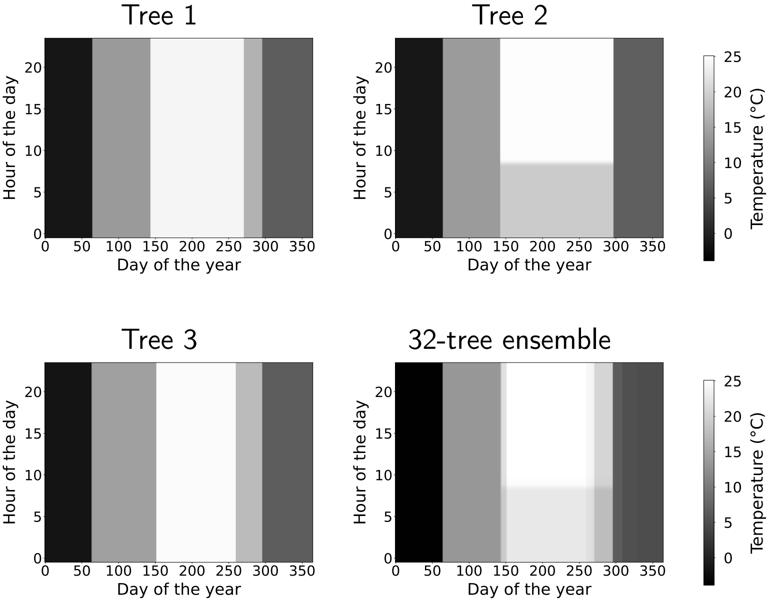
2. Random forests: Average *randomized* trees trained on resampled datasets obtained via bootstrapping

3. Boosting: Combine *complementary* trees that fit residuals of previous trees (scaled down to avoid overfitting)

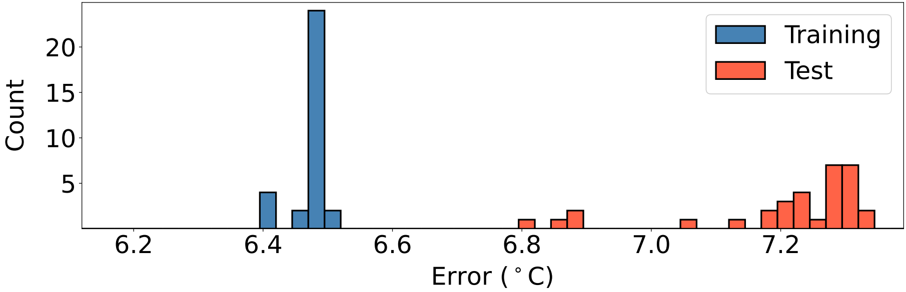
Bootstrapping by sampling from training data



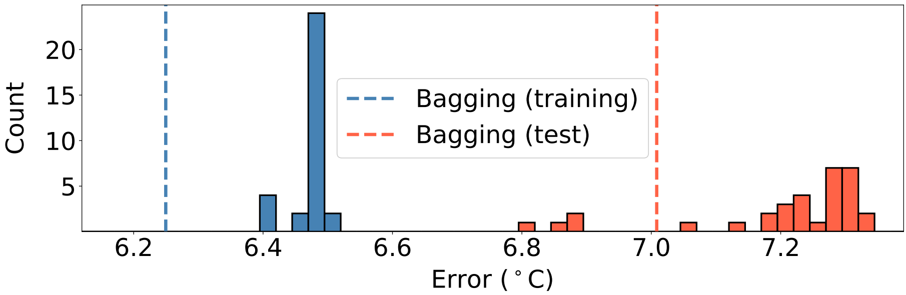
5-leaf trees


Tree 1
Tree 2
Tree 3
32-tree ensemble
Hour of the day
Day of the year
0
5
10
15
20
50
100
150
200
250
300
350
Temperature (°C)
25

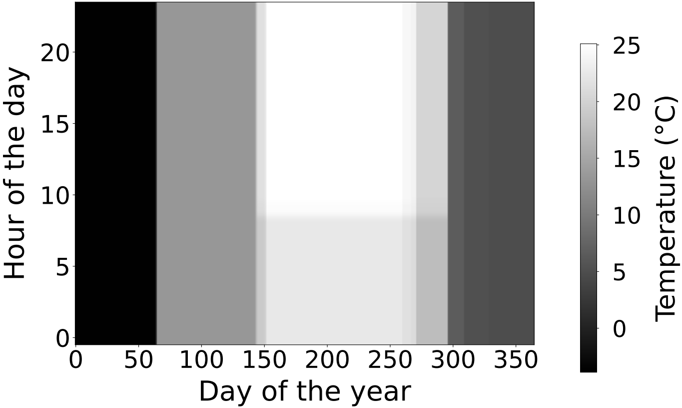
5-leaf trees



5-leaf trees

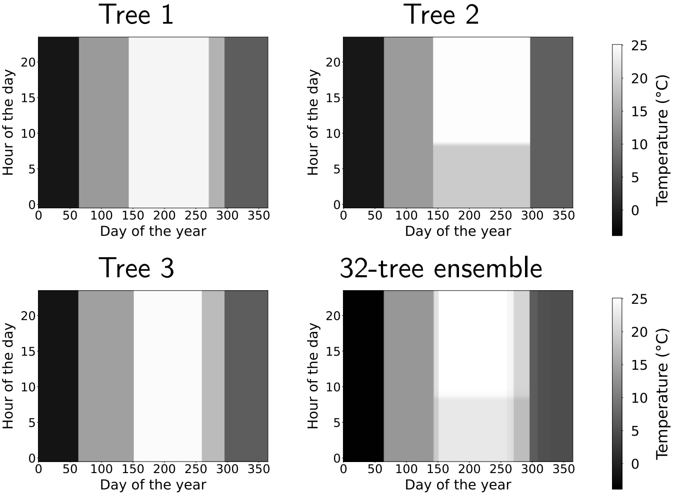

Bagging (training)
Bagging (test)
Count
20
15
10
5
6.2
6.4
6.6
6.8
7.0
7.2
Error (°C)

Less error, but...

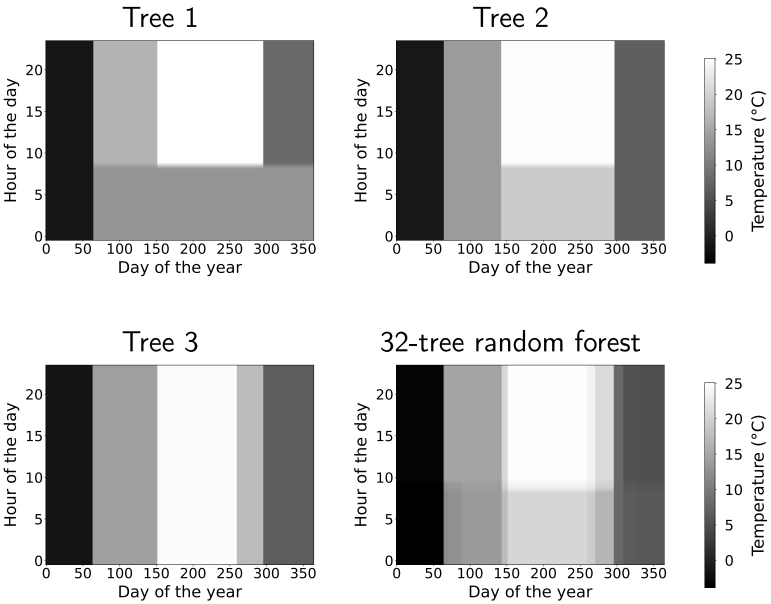


No longer interpretable!

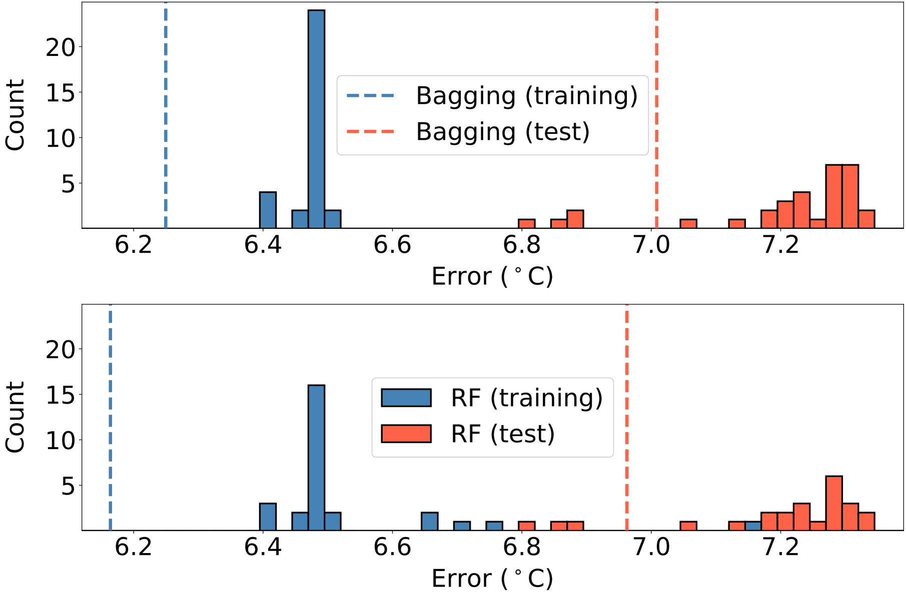
Bagging averages are all similar



Randomized 5-leaf trees



5-leaf trees



Count
20
15
10
5
Bagging (training)
Bagging (test)
6.2
6.4
6.6
6.8
7.0
7.2
Error ($^\circ$C)
Count
20
15
10
5
RF (training)
RF (test)
6.2
6.4
6.6
6.8
7.0
7.2
Error ($^\circ$C)

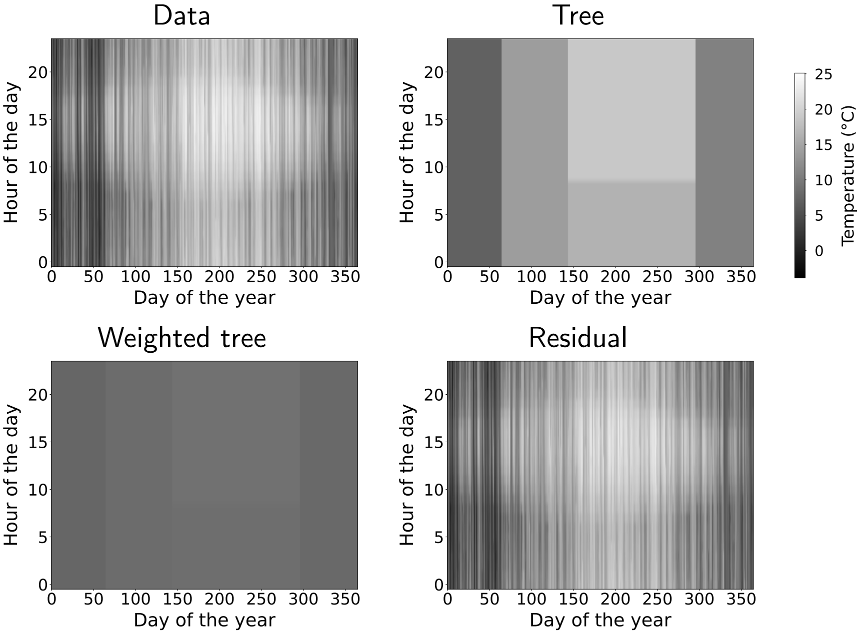
Boosting

Bagging and random forests combine trees trained independently

Boosting combines complementary trees

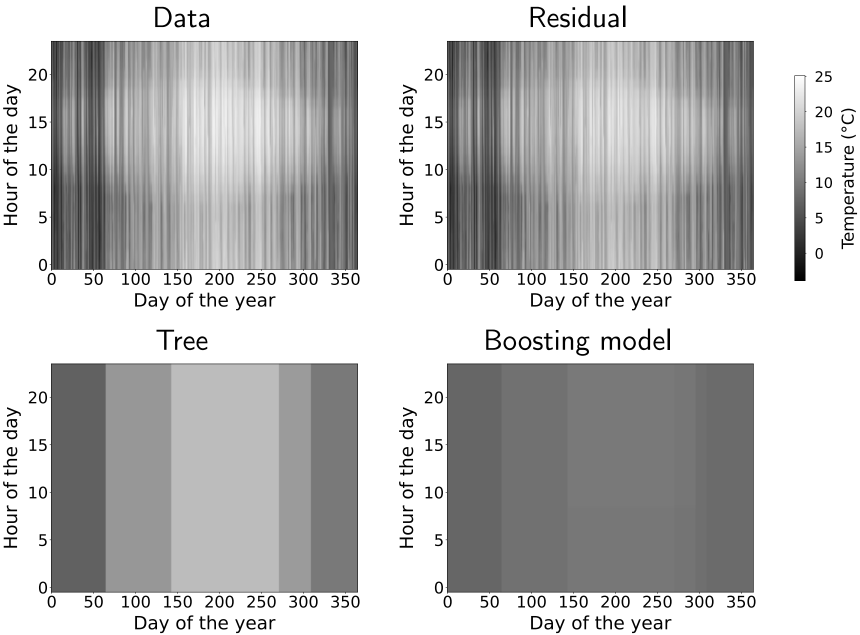
Individual trees are scaled down to avoid overfitting

Boosting



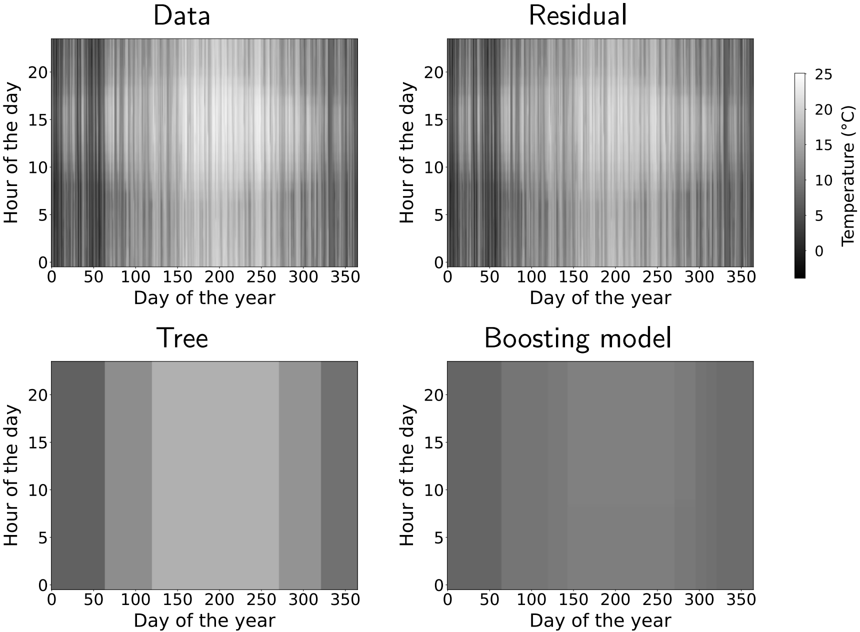
Data
Tree
Weighted tree
Residual
Hour of the day
Day of the year
0
5
10
15
20
0
50
100
150
200
250
300
350
Temperature (°C)
25

Tree 2

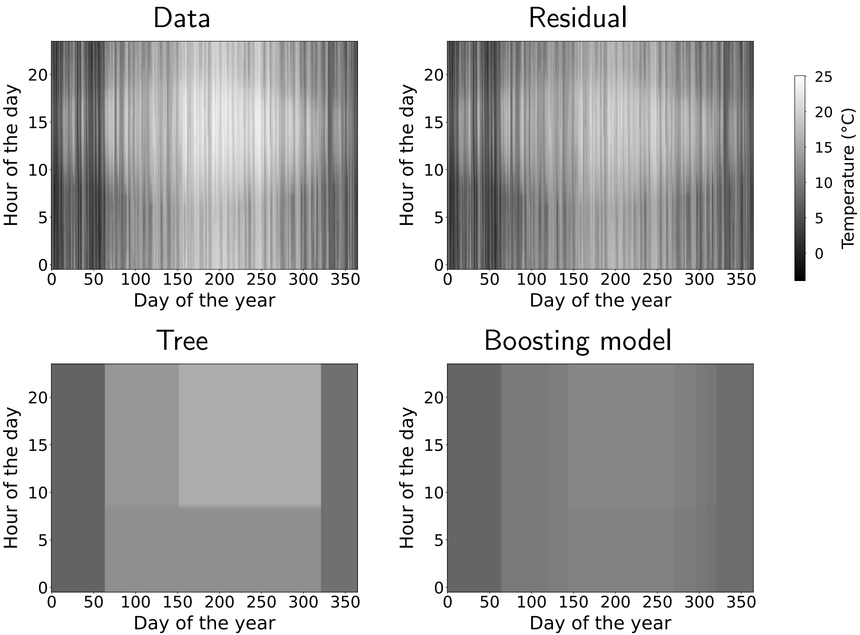


Data
Residual
Tree
Boosting model
Hour of the day
Day of the year
0
5
10
15
20
50
100
150
200
250
300
350
Temperature (°C)
25

Tree 3

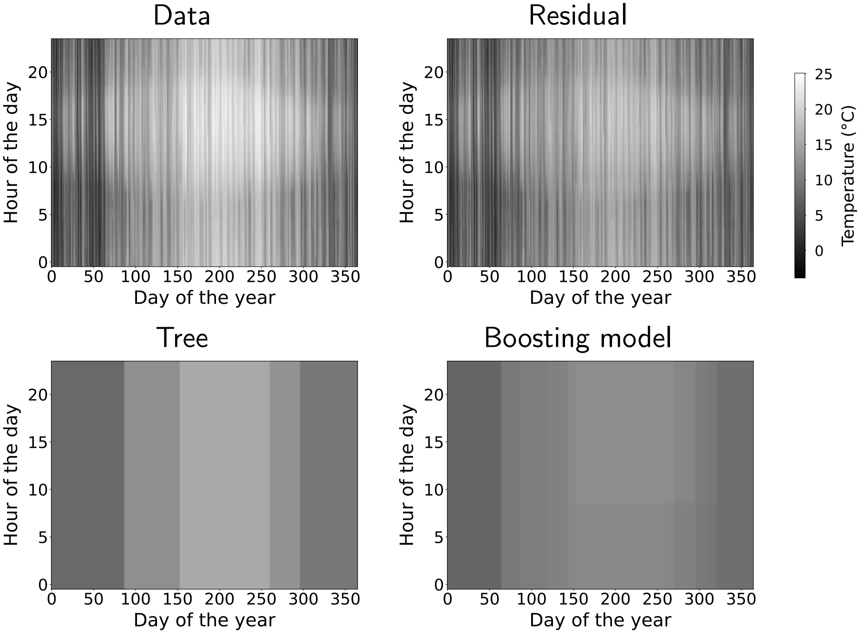

Data
Residual
Tree
Boosting model
Hour of the day
Day of the year
Temperature (°C)

Tree 4

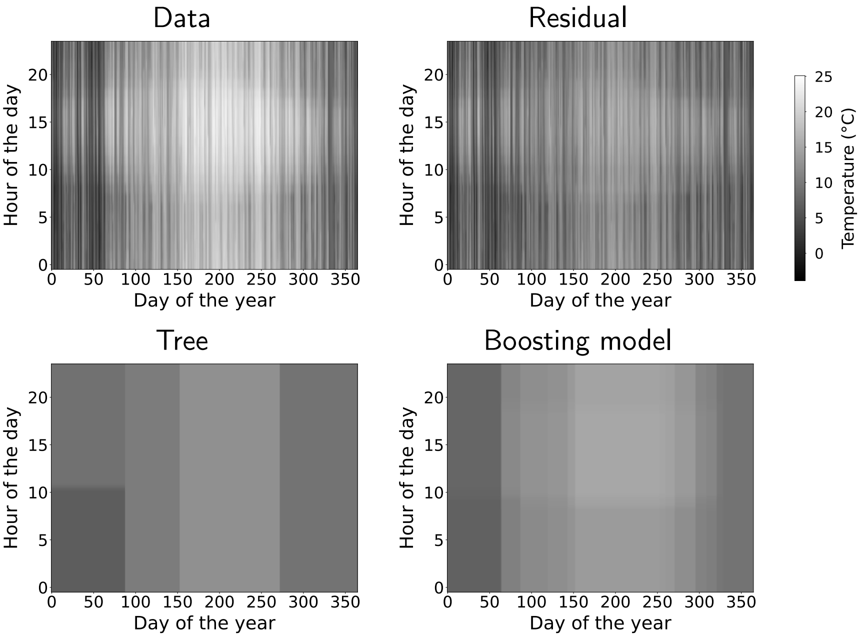


Data
Residual
Tree
Boosting model
Hour of the day
Day of the year
0
5
10
15
20
0
50
100
150
200
250
300
350
Temperature (°C)
25
20
15
10
5
0

Tree 5

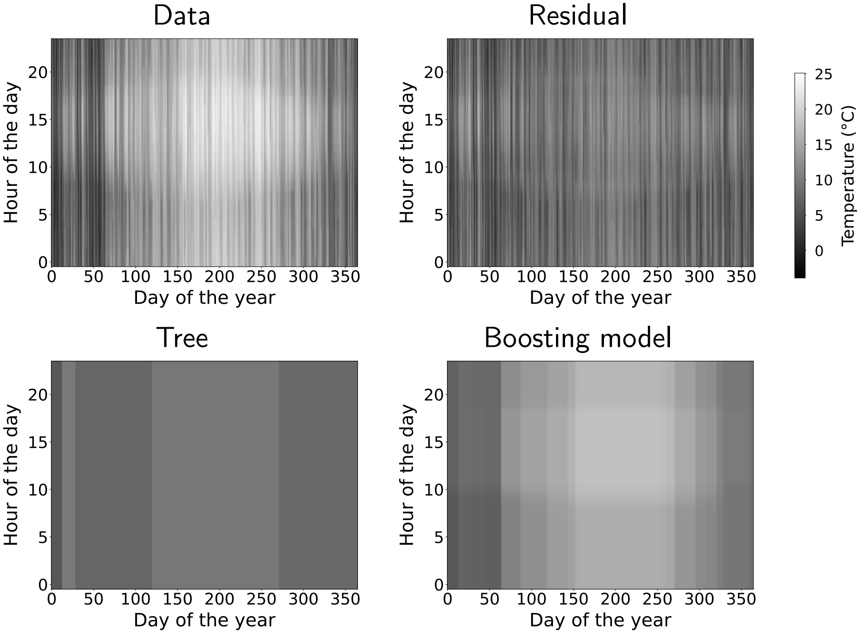

Data
Residual
Tree
Boosting model
Hour of the day
Day of the year
Temperature (°C)
0
5
10
15
20
25
50
100
150
200
250
300
350

Tree 10



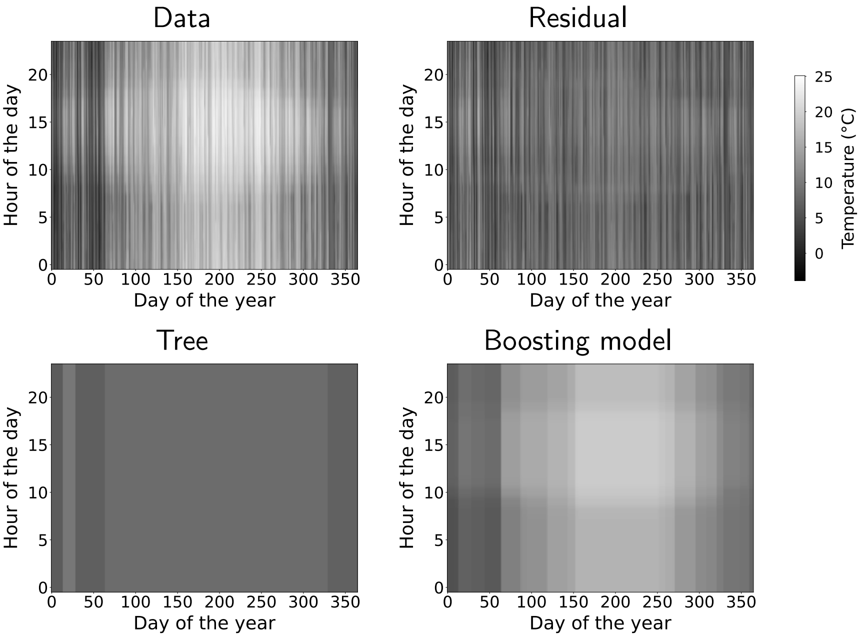
Data
Residual
Tree
Boosting model
Hour of the day
Day of the year
Temperature (°C)
0
5
10
15
20
25
50
100
150
200
250
300
350

Tree 20



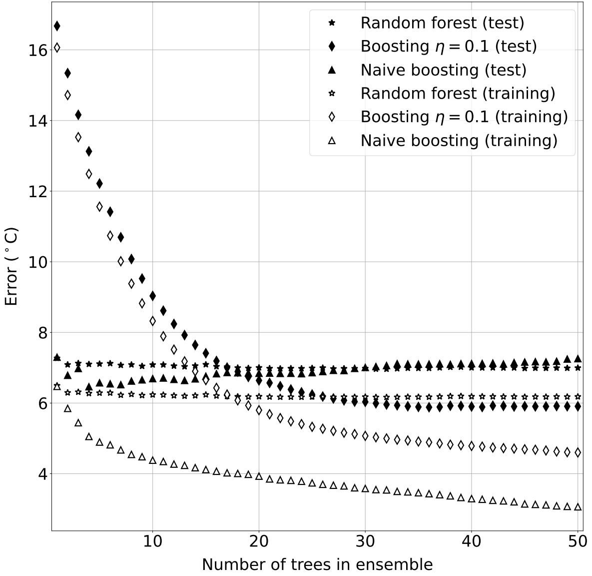
Data
Residual
Tree
Boosting model
Hour of the day
Day of the year
0
5
10
15
20
0
50
100
150
200
250
300
350
Temperature (°C)
25

Tree 30



Data
Residual
Tree
Boosting model
Hour of the day
Day of the year
0
5
10
15
20
0
50
100
150
200
250
300
350
Temperature (°C)
25
20
15
10
5
0

Bagging vs random forests vs boosting

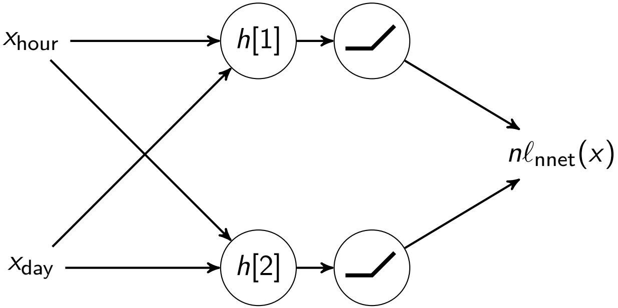


Neural network

Nonlinear function implemented by interleaving

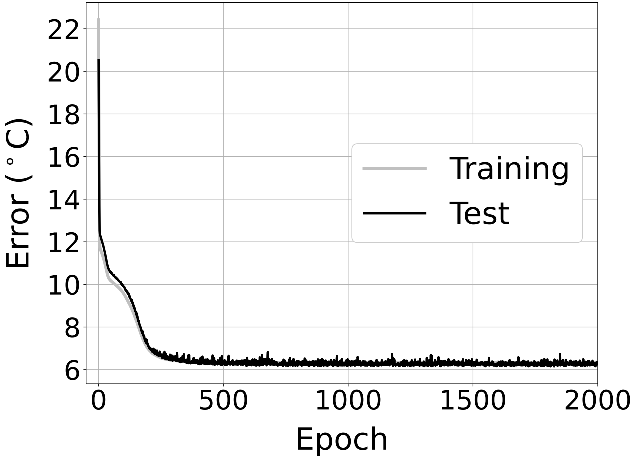
- Linear (affine) transformations
- Nonlinearity

2-layer network for temperature estimation

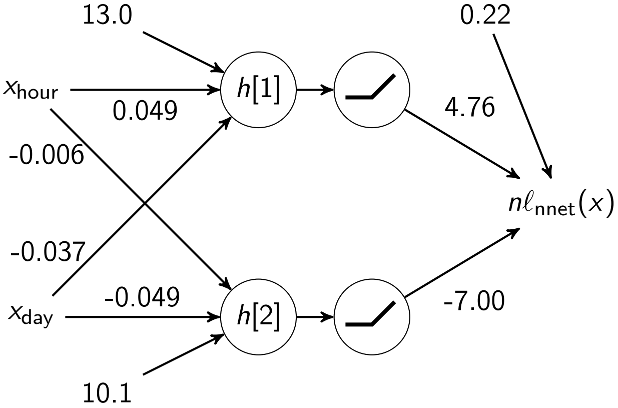


How to estimate network parameters?

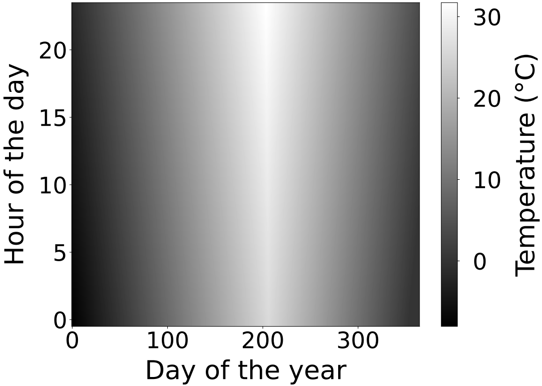
Minimize RSS on training data (separated into batches) via stochastic gradient descent



2-layer network for temperature estimation



Temperature estimation



Training error: 6.32°C Test error: 6.25°C

How can we improve the model

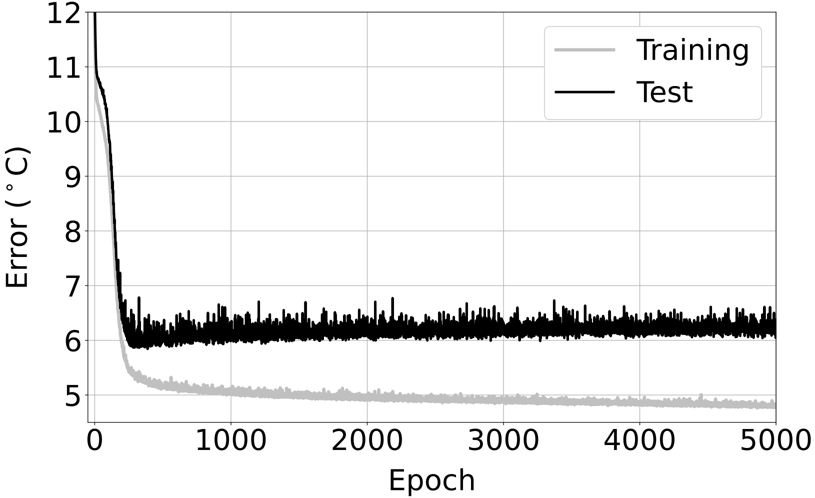
Make network larger and deeper!

4-layer network with 100 hidden variables in intermediate layers

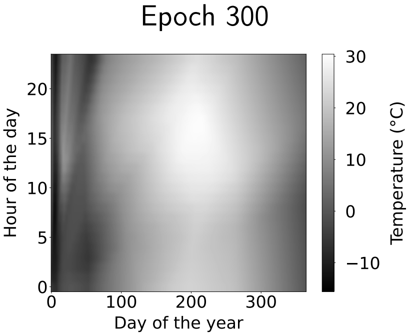
Number of parameters: 20,601

Number of data: 8,760!

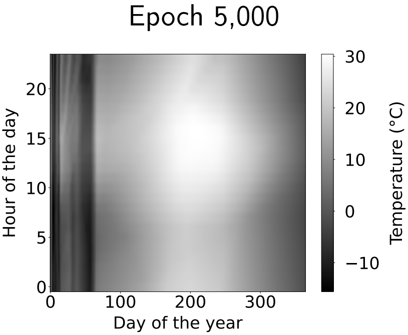
But what about overfitting?



Early stopping mitigates overfitting



Training error: 5.30°C
Test error: 6.06°C



Training error: 4.78°C
Test error: 6.25°C

Different strategies to perform regression

- Linear models:
  - Ordinary least squares
  - Ridge regression
  - The lasso
- Nonlinear models:
  - Regression trees
  - Tree ensembles: bagging / random forests / boosting
  - Neural networks