Department of COMPUTER SCIENCE

WARWICK
THE UNIVERSITY OF WARWICK

Machine Learning
CS342

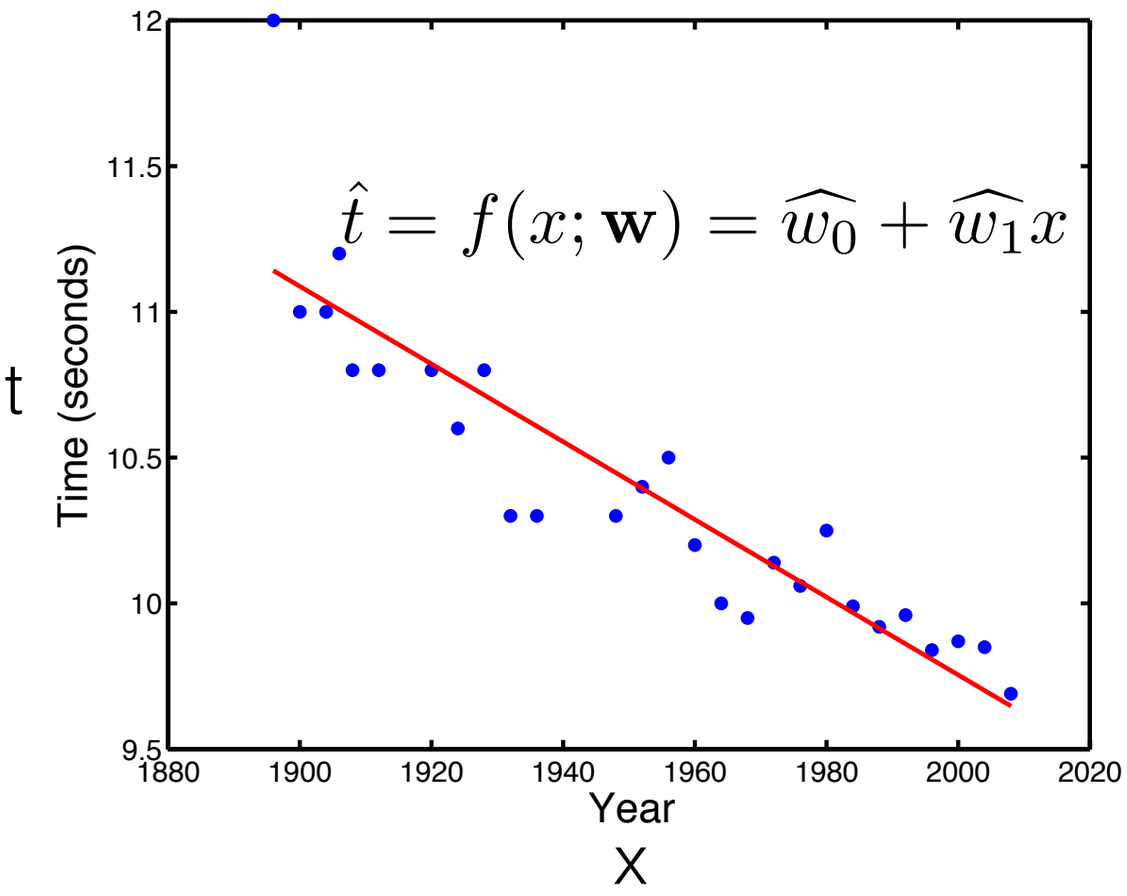
Lecture 3: Linear Regression (OLS): Generalisation, Overfitting, and Validation

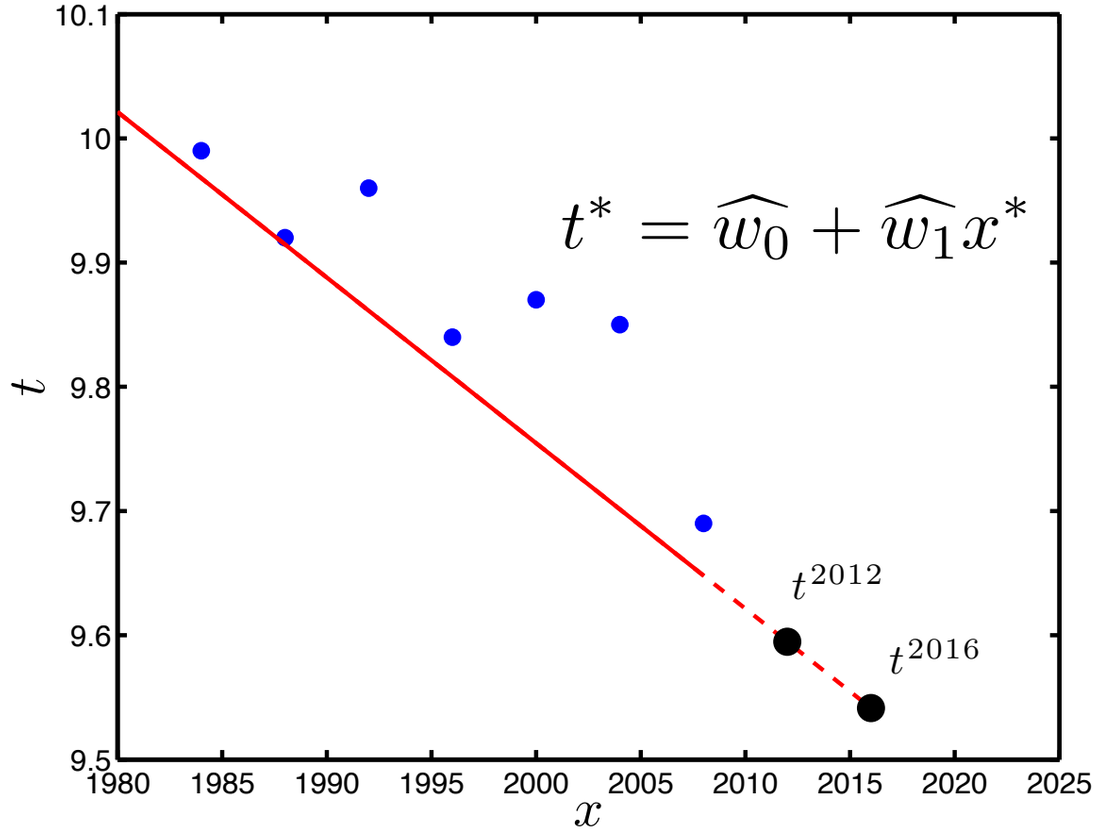
Dr. Theo Damoulas
T.Damoulas@warwick.ac.uk

Office hours: Mon & Fri 10-11am @ CS 307

Recap: OLS

The Ordinary Least Squares (OLS) fit to our training data



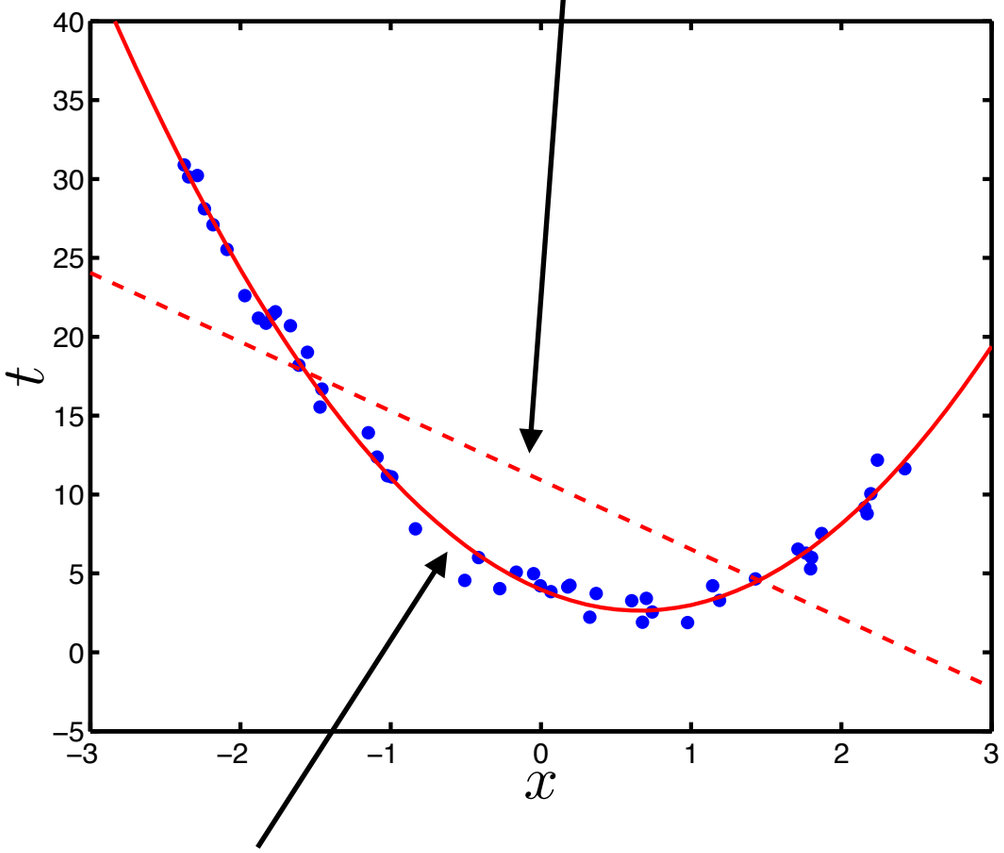


OLS: Squared Loss $$\widehat{w_0}, \widehat{w_1} \leftarrow \underset{w_0, w_1}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} (t_n - f(x_n; w_0, w_1))^2$$

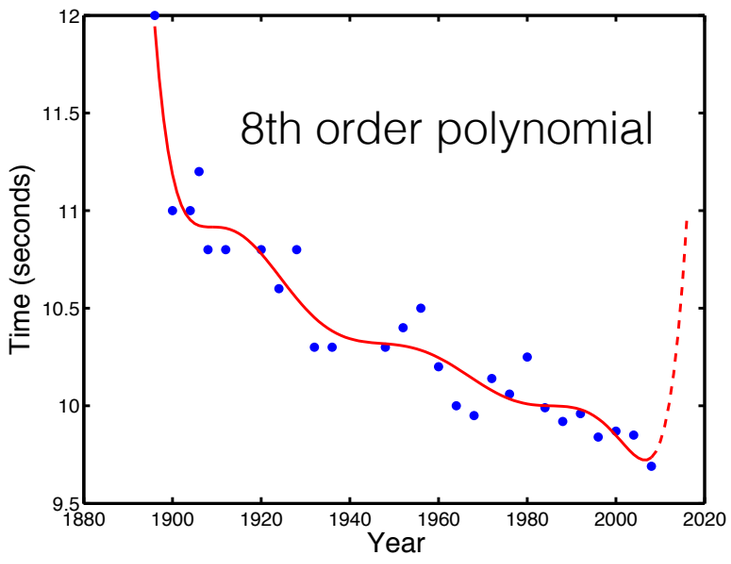
Department of COMPUTER SCIENCE

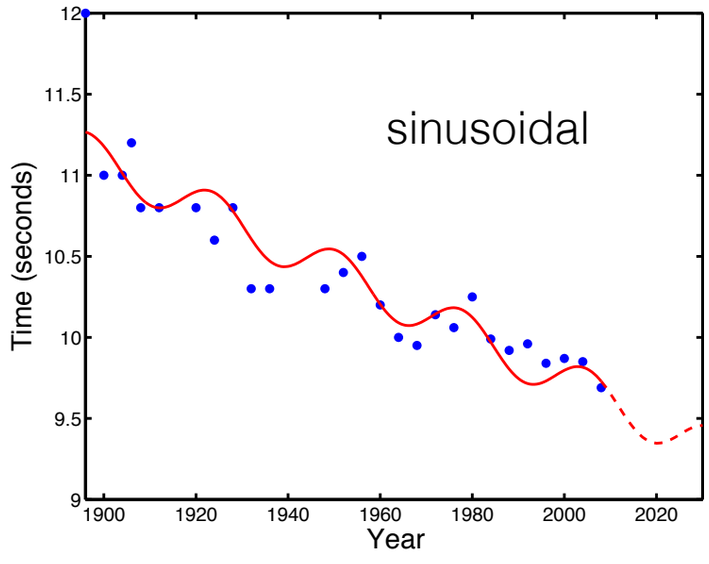
Recap: Non-linear response from linear models

$$f(x; \mathbf{w}) = w_0 + w_1 x$$



$$\mathbf{X} = \begin{bmatrix} x_1^0 & x_1^1 & x_1^2 & \cdots & x_1^K \\ x_2^0 & x_2^1 & x_2^2 & \cdots & x_2^K \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_N^0 & x_N^1 & x_N^2 & \cdots & x_N^K \end{bmatrix}$$





$$f(x; \mathbf{w}) = w_0 + w_1 x + w_2 x^2$$

Still a linear structure (hyper-plane) but in a higher dimensional space

OLS algorithm

"batch mode"

Training algorithm

- Input training data **X**,**t**
- Pre-process **X** (e.g. normalisation)
- Compute OLS parameters by:

$$\widehat{\mathbf{w}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$$

Prediction algorithm

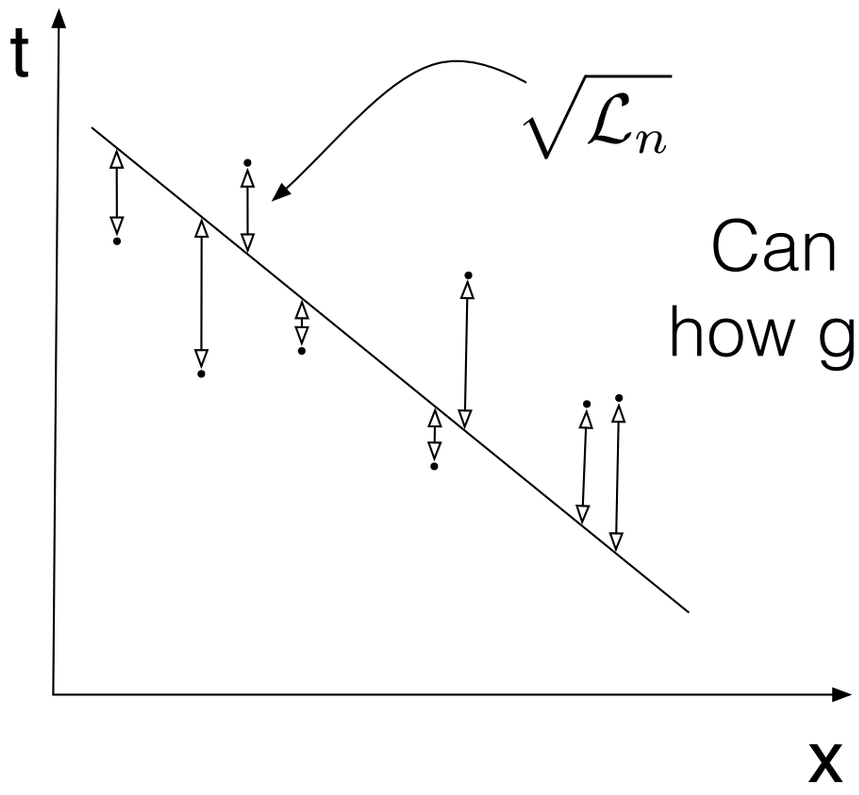
- Input new unseen observations $\mathbf{X}^*$
- Pre-process $\mathbf{X}^*$as in training (e.g. normalisation)
- Compute prediction by:

$$\mathbf{t}^* = \mathbf{X}^*\widehat{\mathbf{w}}$$

$$O(D^2N) \quad \text{Dominant term if N} > \text{D}$$

Generalisation to unseen data

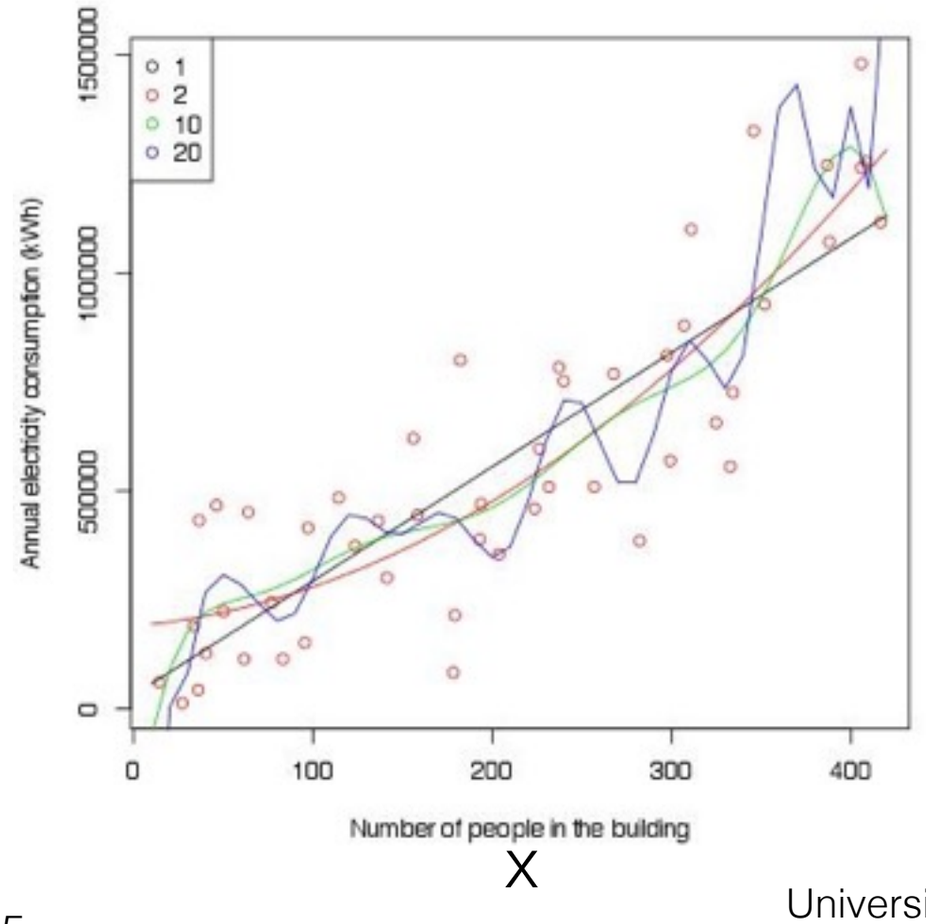
$$\mathcal{L}_n = (t_n - \hat{t}_n)^2 = (t_n - f(x_n; \mathbf{w}))^2 = (t_n - (\widehat{w_0} + \widehat{w_1} x_n))^2$$



Can I use the Loss **on the training data** to describe how good my model will be in predicting unseen data?

Which model is better?

different model complexity (hypothesis spaces)



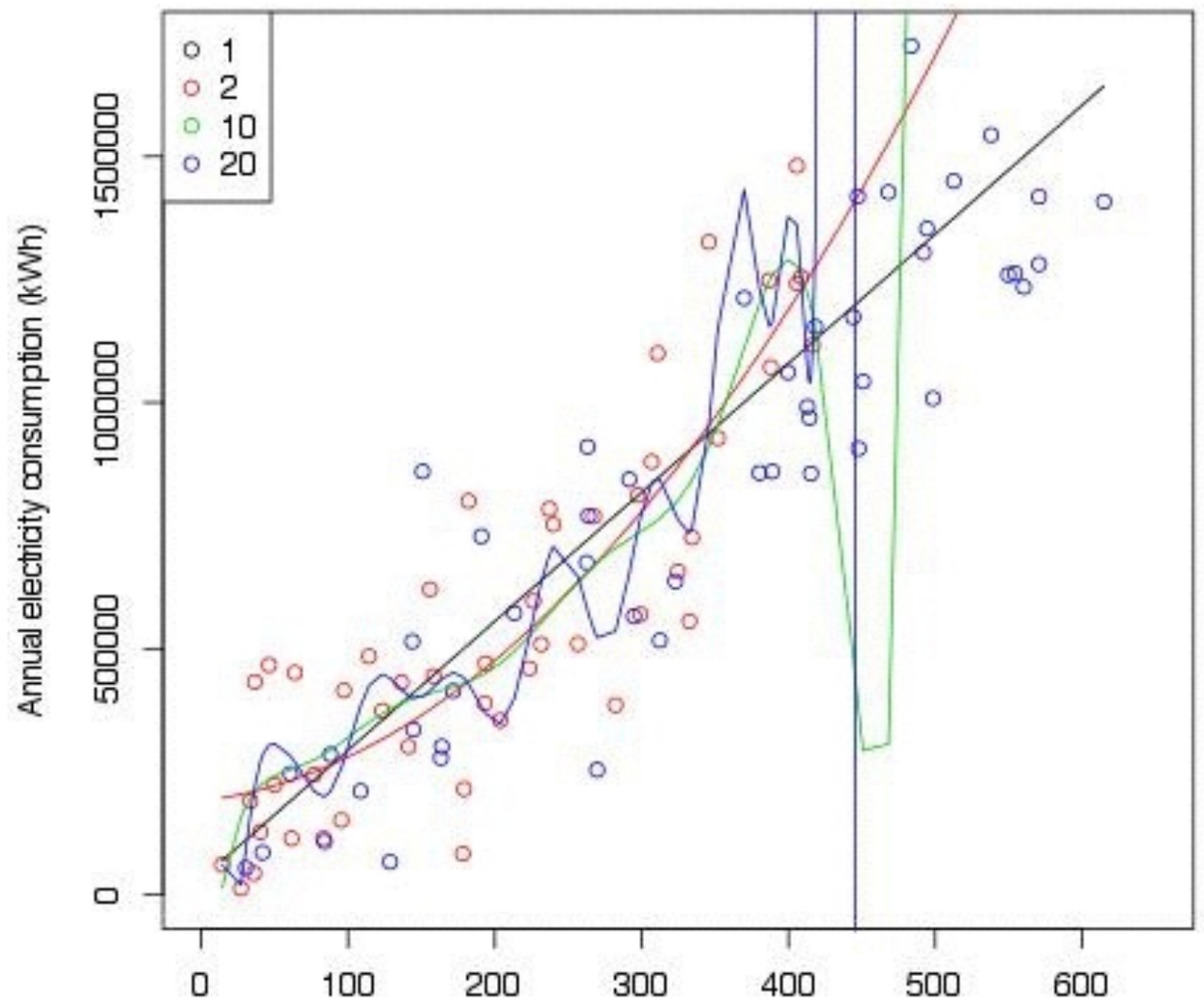
Generalisation: Validation set

Validation set:
A portion of the dataset (a 1/3?) not used during training

Lets predict on validation set

Which model "generalises" better to new/unseen data?

Which model(s) over-fit?

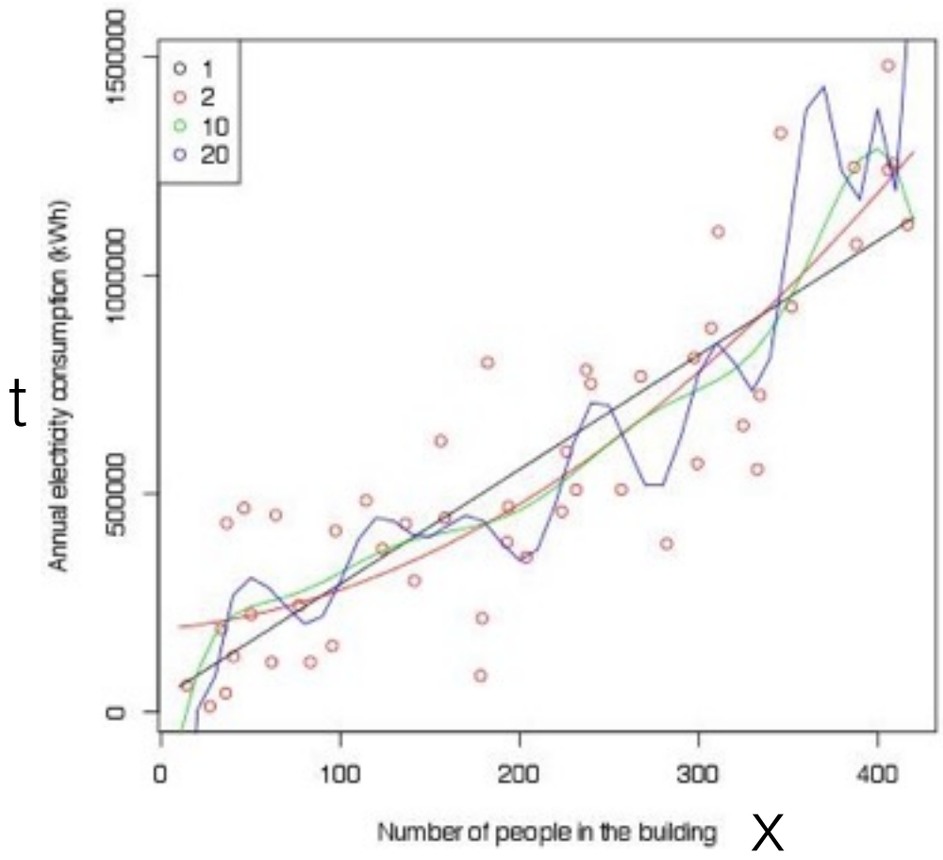


Overfitting:
When we learn a more complex hypothesis then what our data really supports. In that case we are really "fitting the noise" in our training data

Training vs Test/Validation error (RMSE)

When we overfit we are "paying too much attention"
to (the noise in) our training data. **Such models will not generalise well!**

t



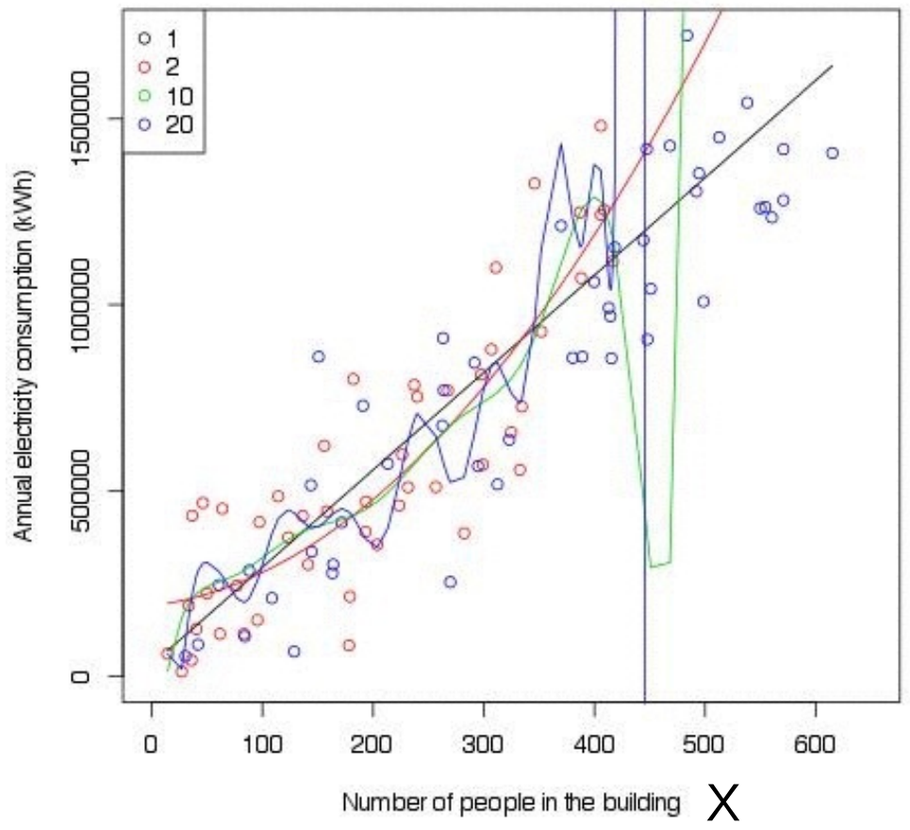
X

1: RMSE=191K
2: RMSE=177K
10: RMSE=170K
20: RMSE=152K

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{n=1}^{N}(t_n - \hat{t}_n)^2}{N}} = \sqrt{\mathcal{L}}$$

"Root Mean Square Error"
The lower the better

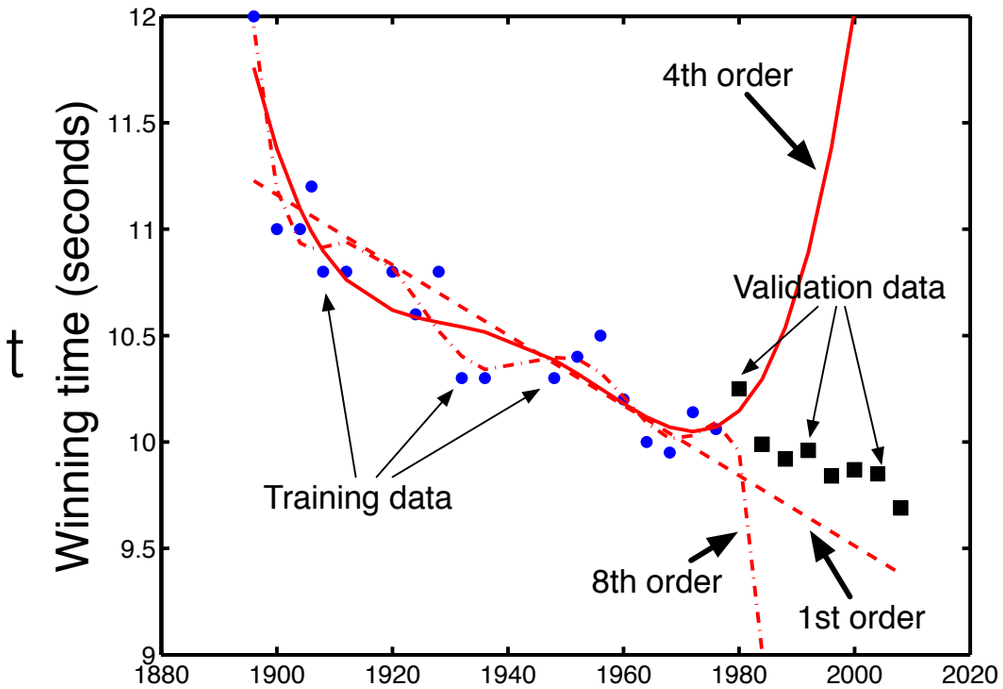
t



X

1: RMSE=200K
2: RMSE=382K
10: RMSE=230M
20: RMSE=5e^14

Training vs Validation error (Olympic data)

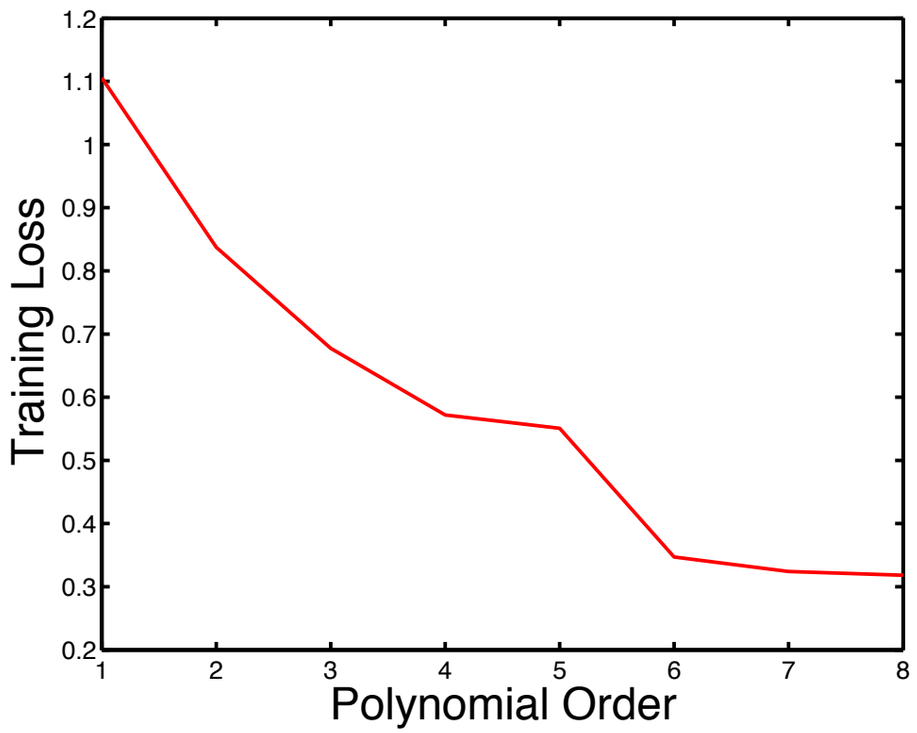


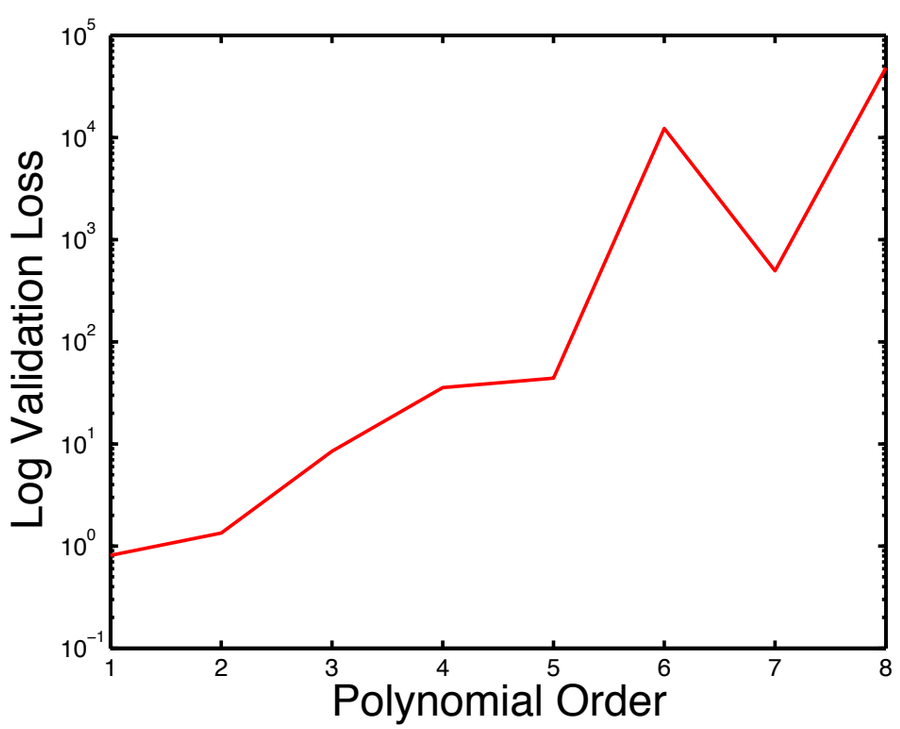
t

Overfitting with overly flexible models: hypothesis not supported by data

0 training error: a very bad idea…

Hand-choosing the validation set: another bad idea





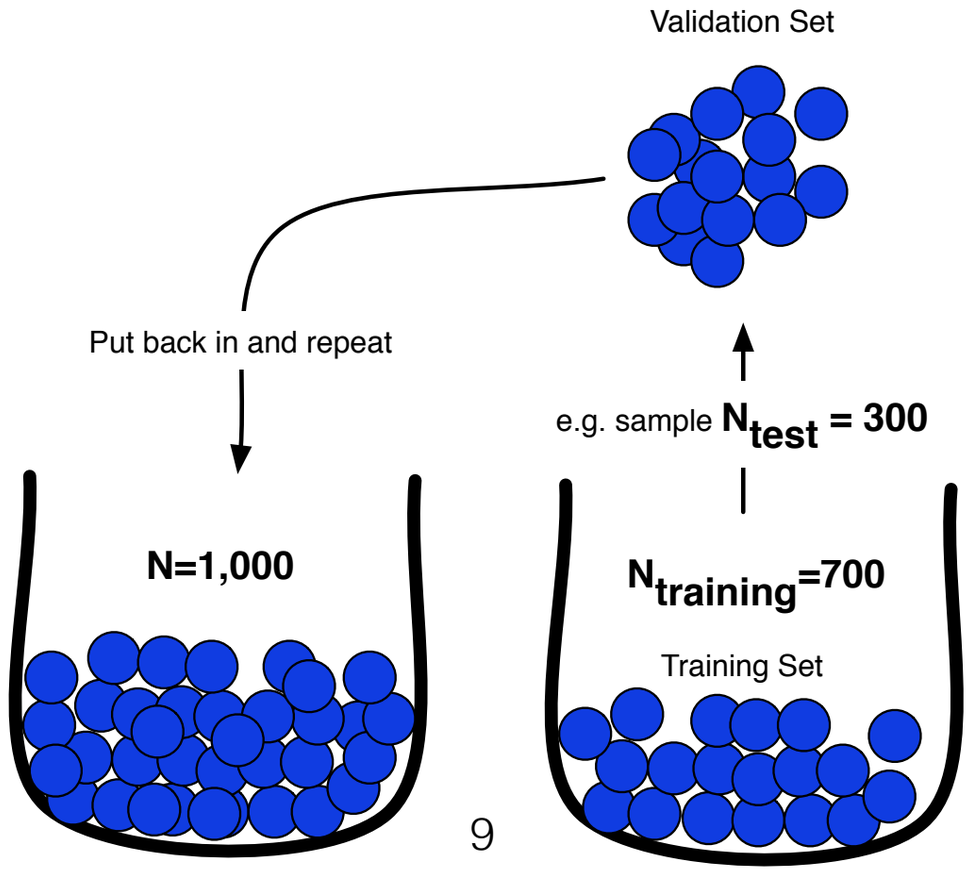
Is this enough?

Not really. You might be slowly hill-climbing/overfitting your validation data!

- Repeated experiments on the same validation set - bad idea
- You become part of the learning algorithm that “sees” the validation set
- Start overfitting the validation set as well

So what can we do?

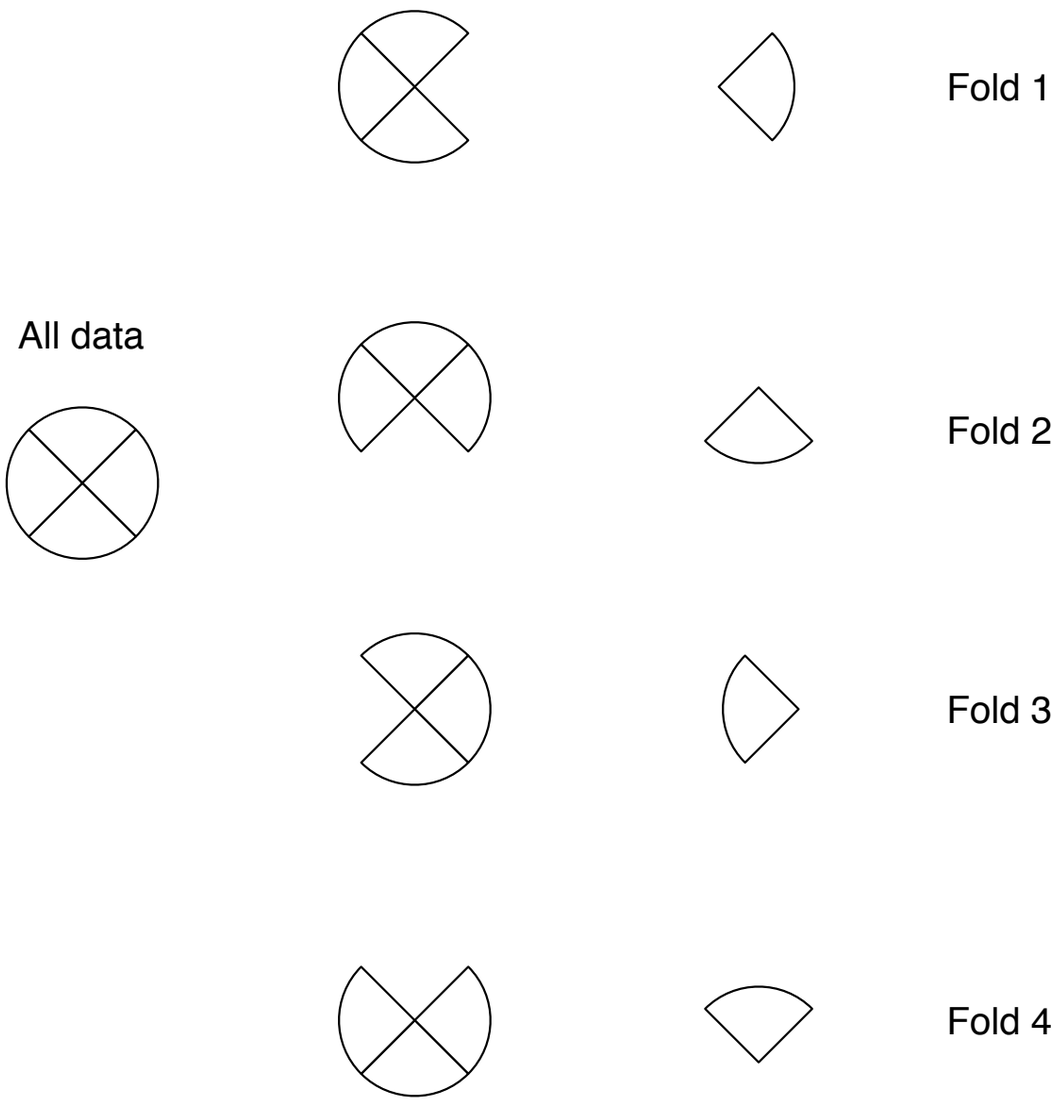
Bootstrap: *Sample with replacement* (the validation set) many times



See also the “Big Data Bootstrap”

Cross-validation

This is 4-fold CV



What is the extreme case?
LOOCV
(computationally expensive)

And we report the **average** performance
e.g. if reporting Loss

$$\mathcal{L}^{\mathrm{CV}} = \frac{1}{N}\sum_{n=1}^{N}(t_n - f(x_n; \widehat{\mathbf{w}}_{-n}))^2$$

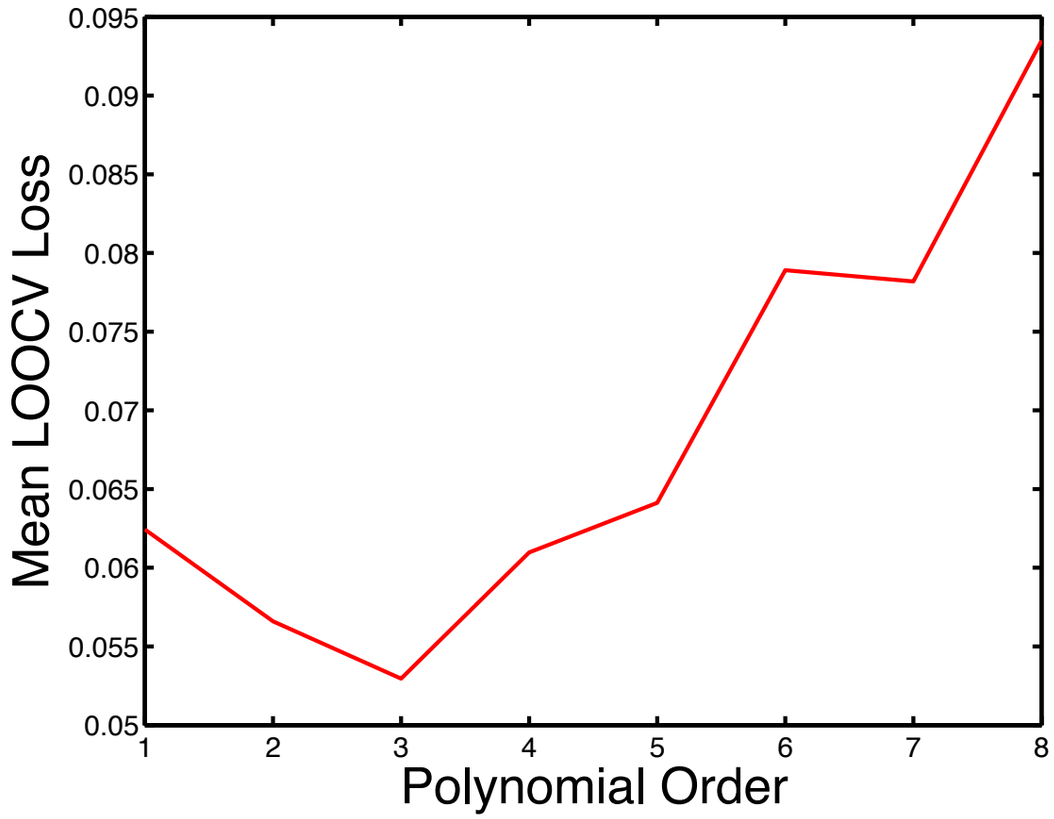
where $\widehat{\mathbf{w}}_{-n}$ is the estimate of the parameters without the n^{th} training example

LOOCV on Olympic data

Wait…3rd order?

What would you choose?

What are the main concerns here?



- Ok, we trust LOOCV more then a single validation set
- Was the validation set chosen at random?
- Is there any significant aspect of the problem that we have ignored?
- What are the main (underlying) assumptions of our learning setting?

What would you do?

Main underlying assumption of many standard SL models

Assumption: ***Data is Independently and Identically Distributed (i.i.d)***

Independent: The N $\{x_n, t_n\}$ samples we have are independently drawn..

- order/sequence of data-points does not matter
- we can reshuffle the N rows of the dataset
- the N rows of dataset are independent
- [careful] x and y are assumed (hopefully they are!) dependent. Some linear or non-linear correlation that we can exploit to learn the mapping

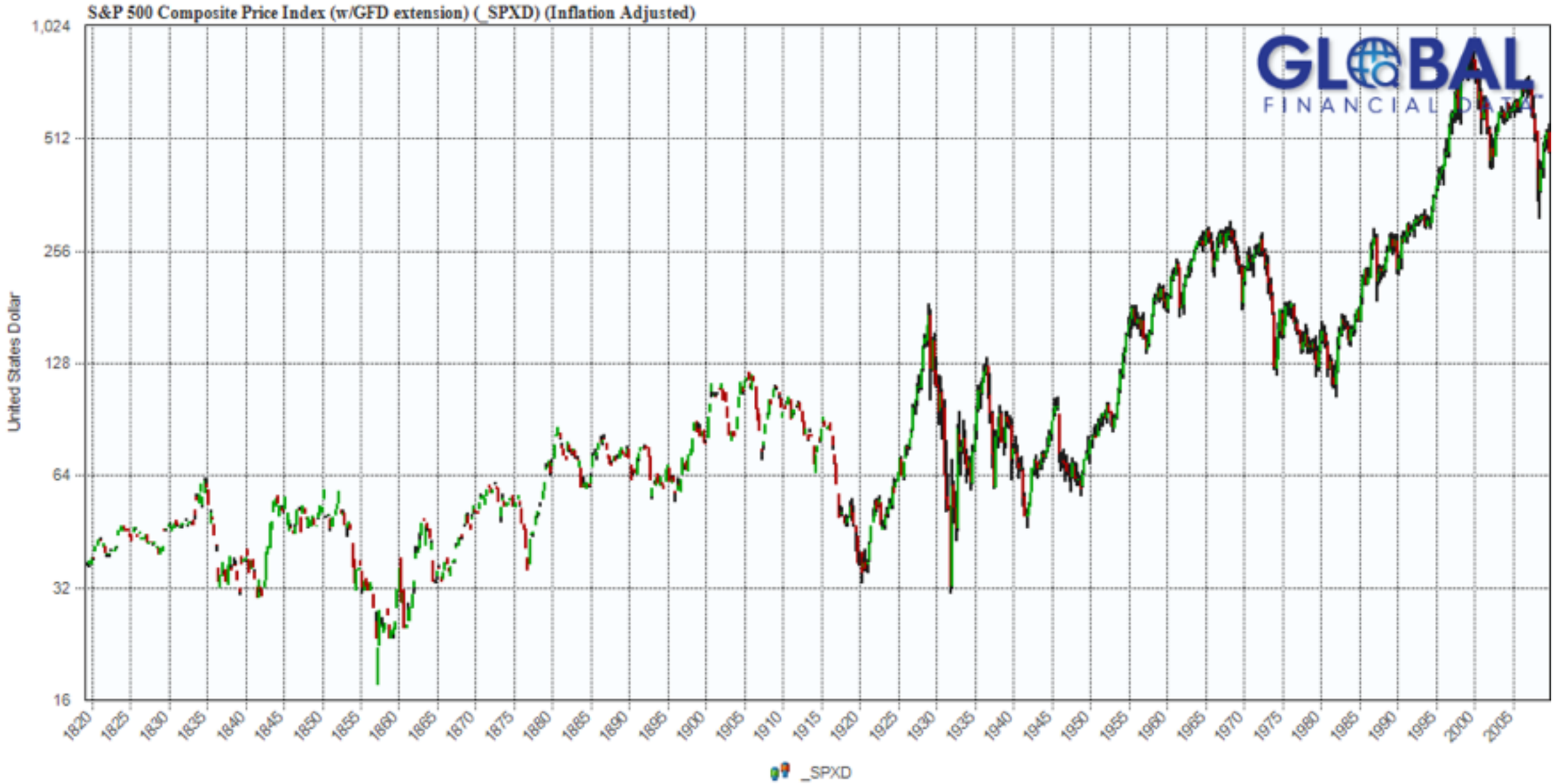
Identical: All of the data (training, validation, test/prediction) are independently drawn ***from the same underlying data generating distribution***

- our model can generalise to unseen data since that data will follow the same phenomena (data generating process)
- assumption of stationarity for the phenomena we study

Department of COMPUTER SCIENCE

Which problems will produce non-i.i.d data?

i.i.d ?



Time (temporal dependence)

Which problems will produce non-i.i.d data?

i.i.d ?

The Last Question by Isaac Asimov © 1956

The last question was asked for the first time, half in jest, on May 21, 2061, at a time when humanity first stepped into the light. The question came about as a result of a five dollar bet over highballs, and it happened this way:

Alexander Adell and Bertram Lupov were two of the faithful attendants of Multivac. As well as any human beings could, they knew what lay behind the cold, clicking, flashing face -- miles and miles of face -- of that giant computer. They had at least a vague notion of the general plan of relays and circuits that had long since grown past the point where any single human could possibly have a firm grasp of the whole.

Multivac was self-adjusting and self-correcting. It had to be, for nothing human could adjust and correct it quickly enough or even adequately enough -- so Adell and Lupov attended the monstrous giant only lightly and superficially, yet as well as any men could. They fed it data, adjusted questions to its needs and translated the answers that were issued. Certainly they, and all others like them, were fully entitled to share In the glory that was Multivac's.

For decades, Multivac had helped design the ships and plot the trajectories that enabled man to reach the Moon, Mars, and Venus, but past that, Earth's poor resources could not support the ships. Too much energy was needed for the long trips. Earth exploited its coal and uranium with increasing efficiency, but there was only so much of both.

Text (sequence dependence)

Which problems will produce non-i.i.d data?

i.i.d ?

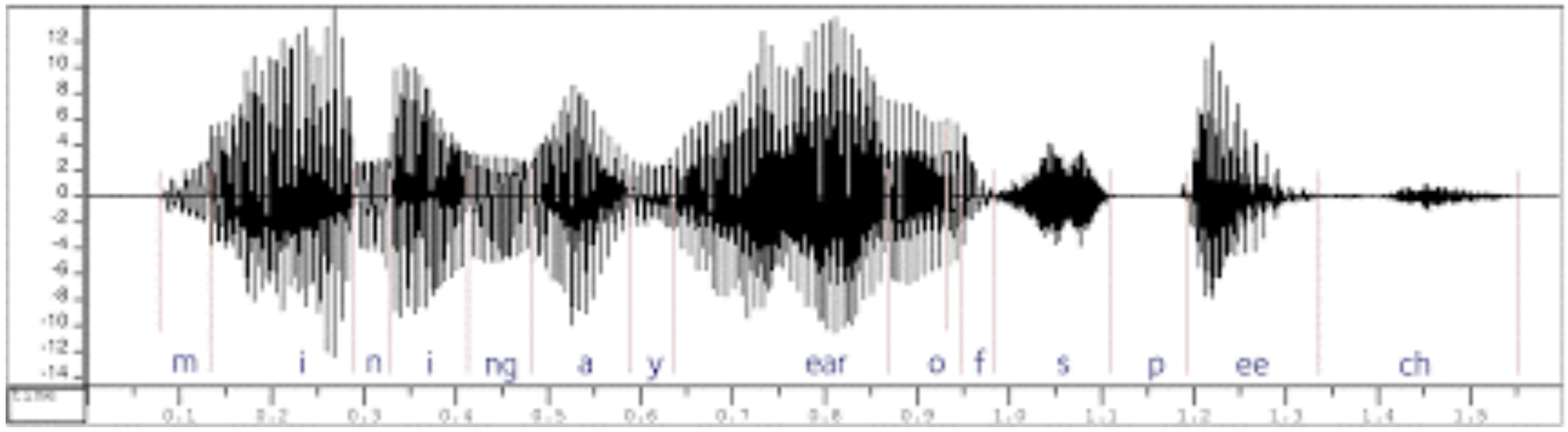
Spatial data (spatial dependence)

Department of COMPUTER SCIENCE

Which problems will produce non-i.i.d data?

Task: Given acoustic examples like this predict gender

i.i.d ?

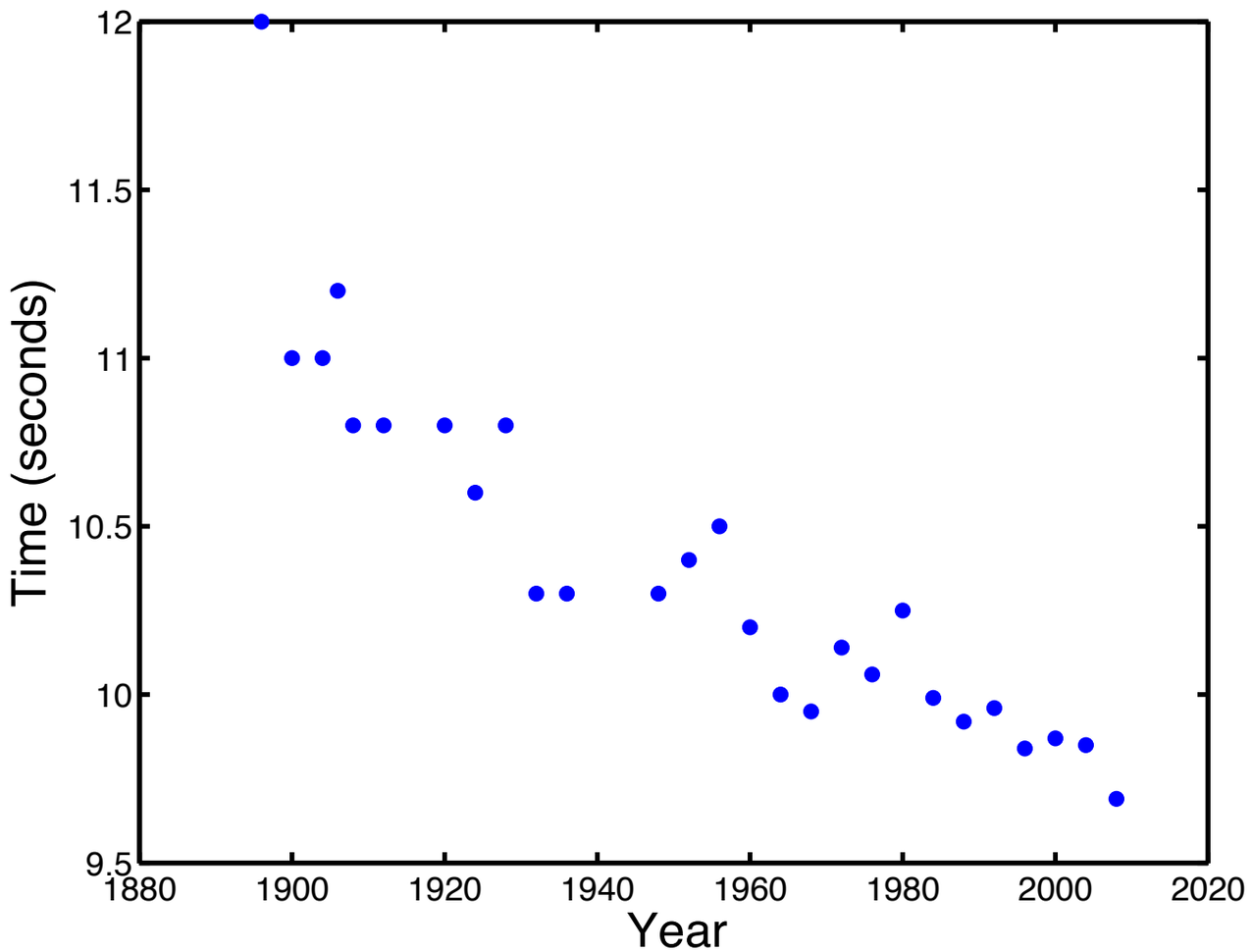


Task: Predict the energy level at a frequency given the time

Depends on the task each time!

How about our Olympic data?

i.i.d ?



Time (temporal dependence?)

Non-i.i.d data

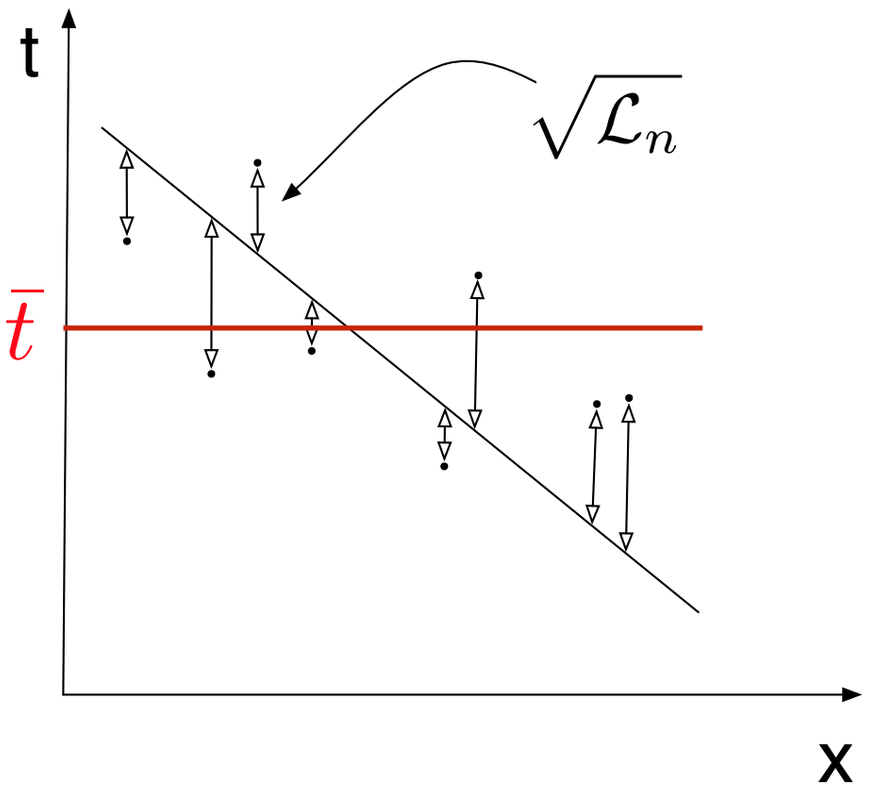
- i.i.d. assumption often invalid
- we exploit the dependence to “borrow strength” from “neighbours”
- be able to identify when severely violated
- some ML models have build in capabilities to capture these dependencies
- including these dependencies in our model will improve performance

Temporal (sequence) dependence:
e.g. Time-series models, Hidden Markov Models, Graphical models

Spatial dependence:
e.g. Spatial statistics, Gaussian Processes, Graphical models

Back to OLS: Some more performance metrics

$$\mathcal{L}_n = (t_n - \hat{t}_n)^2 = (t_n - f(x_n; \mathbf{w}))^2 = (t_n - (\widehat{w_0} + \widehat{w_1} x_n))^2$$



$$\text{RMSE} = \sqrt{\frac{\sum_{n=1}^{N}(t_n - \hat{t}_n)^2}{N}} = \sqrt{\mathcal{L}}$$ ↓

Pearson *sample* correlation coefficient:

$$r(\mathbf{t}, \hat{\mathbf{t}}) = \sqrt{\frac{\sum_{n=1}^{N}(\hat{t}_n - \bar{t})^2}{\sum_{n=1}^{N}(t_n - \bar{t})^2}}$$ ↑

%variation explained

Coefficient of determination ("R square")

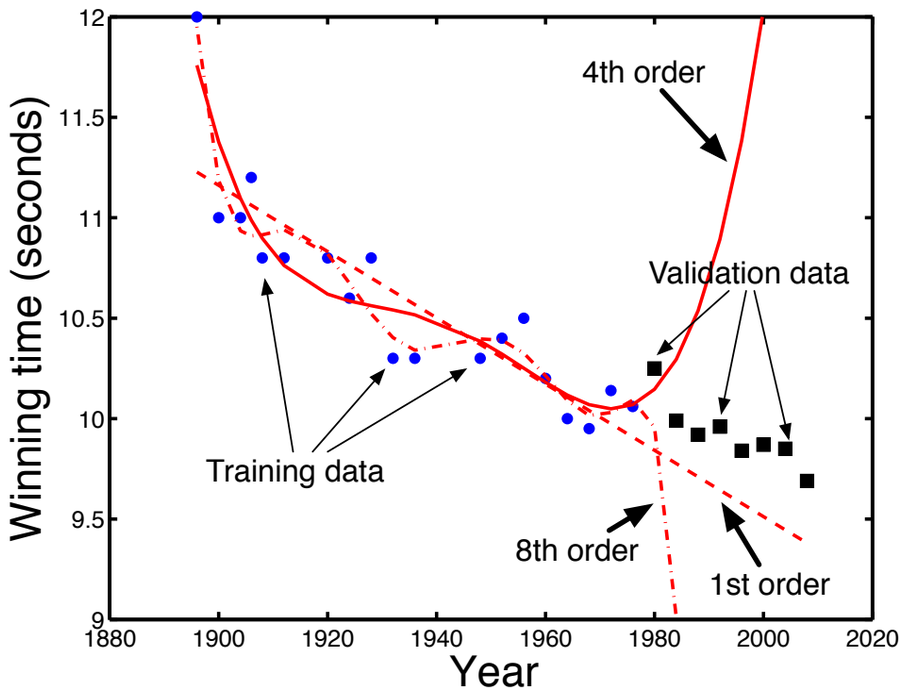
$$R^2 = 1 - \frac{\sum_{n=1}^{N}(t_n - \hat{t}_n)^2}{\sum_{n=1}^{N}(t_n - \bar{t})^2}$$ ↑

Curse of dimensionality

- OLS is prone to overfitting in high-D as we introduce more parameters, one for every dimension.

- This implies more Degrees of Freedom (DoF) and a more complex model

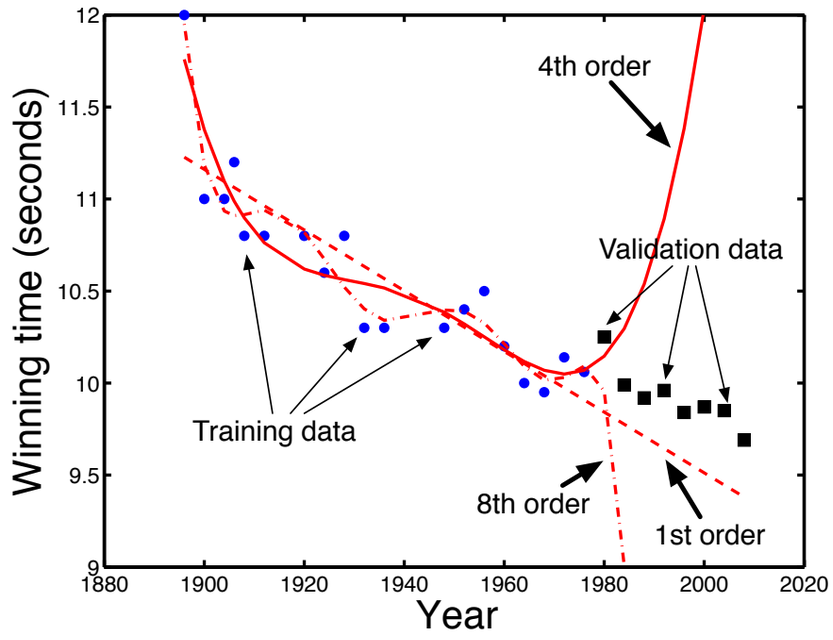
- Curse of dimensionality for OLS: "*As we increase the complexity of our model (e.g. introduce high order polynomial expansions), we increase the DoF or number of parameters and therefore we require even more observations/inputs N to fit the model.*"



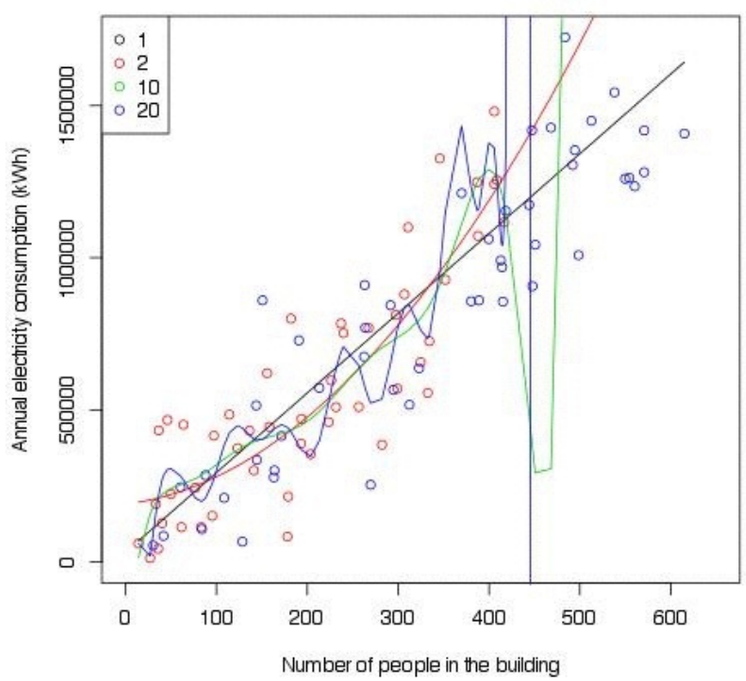
Department of COMPUTER SCIENCE

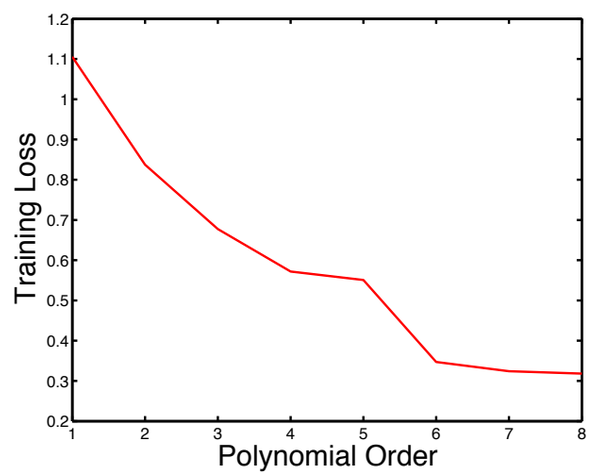
Summary so far

Generalisation & Validation data

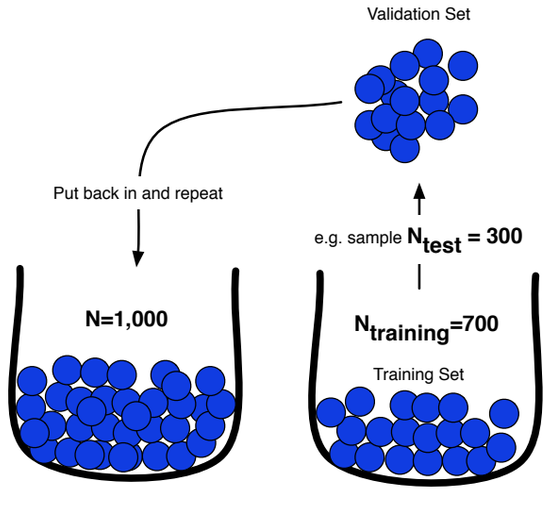


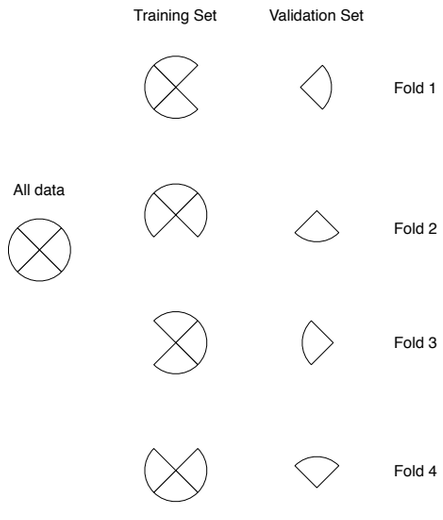
Overfitting and Curse of Dimensionality



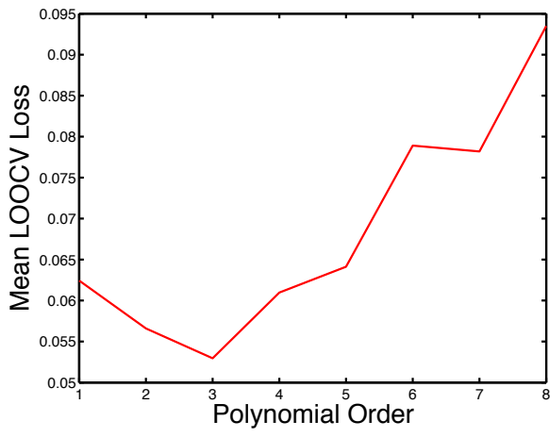


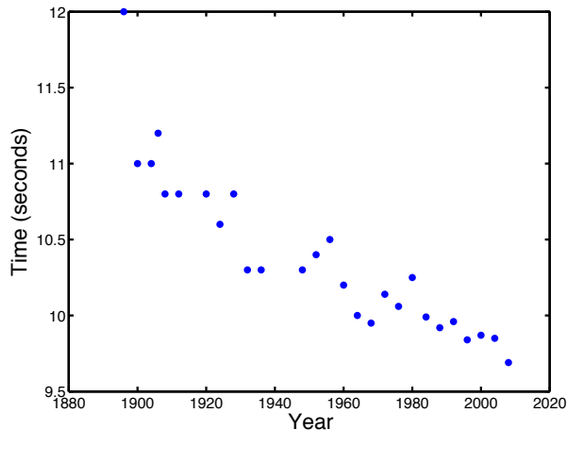
Bootstrap & Cross-validation





Model Selection and IID assumption





What else can we do for CoD/Overfitting and Interpretation?

We want to have many attributes or expand our input space to a new feature space of higher dimensions. But CoD says you need even more observations to do so in order to avoid overfitting

$$f(x; \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + w_5 x^5$$

If all w's are zero simplest model that always outputs 0.
What if only w_0 non-zero?

As w's depart from zero one by one, our model becomes more complex
(0, a constant, a line, a 2-D plane, 3-D hyper-plane, etc..)

Use parameter values as strength/indication of correlation? ***Interpretation***

Regularisation and Penalised Least Squares (PLS)

Regularisation: We control (regularise/penalise) the complexity of the linear model by restricting the "magnitude" (e.g. the sum of the absolute values) of our parameters.

Can also think of it is ***coupling*** together our parameters to restrict their magnitudes, or as a "***competition***" between themselves.

One way: keep this value low $\sum_d w_d^2 = \mathbf{w}^{\mathrm{T}}\mathbf{w}$

Update our Loss function to include it $\mathcal{L}' = \mathcal{L} + \lambda \mathbf{w}^{\mathrm{T}}\mathbf{w}$

Repeat derivation for new solution (PLS)

$$\widehat{\mathbf{w}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + N\lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$$

Generalisation-Overfitting Trade-off

6 datapoints, 5-order polynomial, varying the strength of regularisation

You can use CV to choose the strength (complexity)