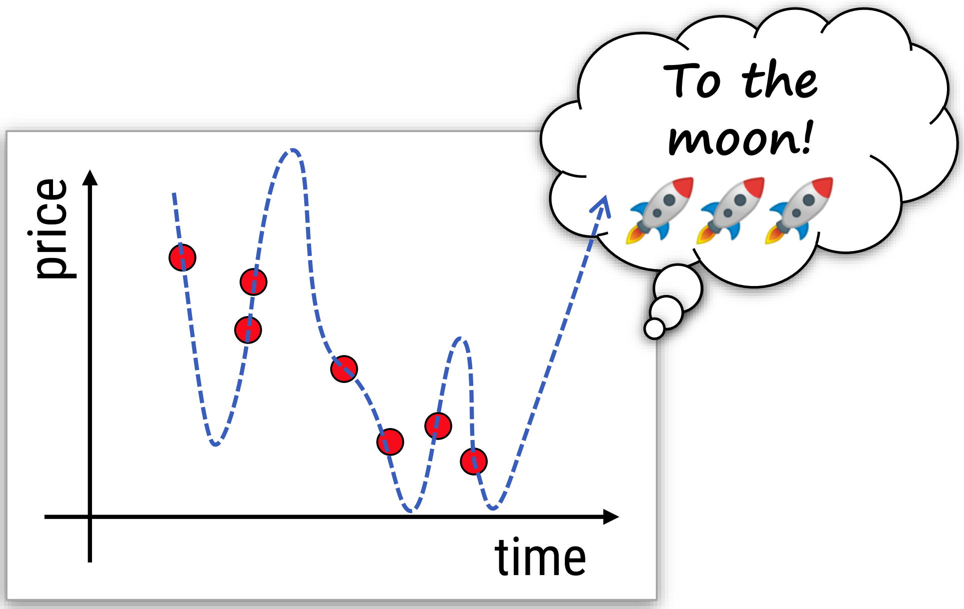
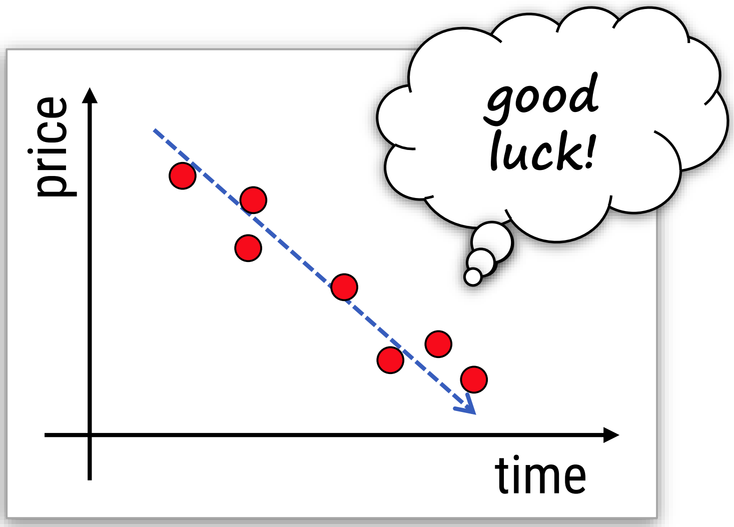
Modelling 2

STATISTICAL DATA MODELLING

Institut für Informatik

JOHANNES GUTENBERG UNIVERSITÄT MAINZ





Chapter 7

Generalization

Michael Wand · Institut für Informatik, JGU Mainz · michael.wand@uni-mainz.de

Video #07

Statistical Learning Theory

- **Limits of learning:** No Free Lunch
- **Frequentist:** Statistical Learning Theory
- **Bayesian Model Selection**

There is…

No Free Lunch

…just somebody else is paying.

Universal Learning Algorithm

Can we find a universal learning algorithm?

- Should works on *any* problem
- With good performance
  - At least better than chance

Counter-question

- Depends on how you define *any*

Strict definition: Really any

- Then: Answer is no.
- “No free lunch theorem” of machine learning

No Free Lunch Theorem

Informal Statement

- Consider machine learning task
  - E.g. classification
  - E.g. regression
- It is impossible to learn models that
  - Perform better than random choice
    if we do not restrict the problem class a priori
  - “No successful learning without priors”
- Two variants / components

(NFL1) All algorithms equal (on average) over all possible problems

(NFL2) Generalization requires *using* prior knowledge

No Free Lunch

Formalization
(for Classification)

No Free Lunch Theorem (1)

Assumption

- No prior information
- All distributions equally likely

Consequence

- All predictors (incl. random choice) are equally good (bad)
- Expected average performance is pure chance

No Free Lunch Theorem (1)

Unknown

- Features $\mathbf{x} \in \Omega(X) = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
- *True* labeling function $y: \Omega(X) \rightarrow \{0,1\}$
- Training Data $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n),\ y_i \coloneqq y(\mathbf{x}_i)$

Complexity

- N possible input features
  - Usually, N is very, very large
- Labels are binary
  - There are 2^N possible labelings

$$\left\{\begin{array}{c} \mathbf{x}_1 \rightarrow \\ \mathbf{x}_2 \rightarrow \\ \mathbf{x}_3 \rightarrow \end{array}\right\} \begin{pmatrix}0\\0\\0\end{pmatrix}\begin{pmatrix}0\\0\\1\end{pmatrix}\begin{pmatrix}0\\1\\0\end{pmatrix}\begin{pmatrix}0\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}1\\1\\0\end{pmatrix}\begin{pmatrix}1\\1\\1\end{pmatrix}$$

No Free Lunch Theorem (1)

Training data

- Training features $X_T = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \Omega(X),\ n < N$
- Training labels $y(\mathbf{x}_i)$ given for all $i = 1 \dots n$

Result of training

- Learned model $h \colon \Omega(X) \to \{0,1\}$ ("Hypothesis")

Problem: Generalization

- Non-training features $X_G = \Omega(X) \backslash X_T$
- We want to infer $y(\mathbf{x})$ for $\mathbf{x} \in X_G$

No Free Lunch Theorem

Quality measure: Generalization error

$$L(h) \coloneqq \frac{1}{\#X_G} \sum_{\mathbf{x} \in X_G} |h(\mathbf{x}) - y(\mathbf{x})|$$

- Average *generalization* error
  - I.e., average on *off-training* data

Assumption

- Draw true labeling function

  $$y: \Omega(X) \rightarrow \{0,1\}$$

  uniformly & randomly from set of all such functions
- (Really) no prior knowledge (possible)

No Free Lunch Theorem

Theorem ("no free lunch (1)")

- Under these assumptions
  - Pick labeling function uniform, randomly from function space
- All possible models h have the same expected performance

$$L(h) = 0.5$$

  - Averaged over all potential true $y \in \{y | y: \Omega(X) \rightarrow \{0,1\}\}$
- Corollary: All ML-algorithms are equally good (here)
  - Includes fancy ones like SVMs, Deep Nets
  - But same for "always answer 0" or random guessing

Proof (NFL 1)

- We have 2^N possible labeling
$$y: \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \to \{0,1\}$$
  - We pick any of this with same probability
- Look at one off-training point $\mathbf{x}_i \notin X_T$
  - There are 2^{N-1} functions with $y(\mathbf{x}_i) = 0$
  - There are 2^{N-1} functions with $y(\mathbf{x}_i) = 1$
- Chance of labeling are 50:50
  - Independent of training data
  - $h(\mathbf{x}_i)$ will be wrong 50% of the time (no matter the choice)
  - $\mathbb{E}_{y:\{\mathbf{x}_1,\dots,\mathbf{x}_N\}\to\{0,1\}}[|h(\mathbf{x}_i) - y(\mathbf{x}_i)|] = 0.5$
- This holds for all off-training points $\Rightarrow L(h) = 0.5$

Conclusion

No Free Lunch (1)

- No universal learning
  - Pure mathematics / perfect symmetry

    no prior knowledge = all functions y are equal
  - Random problems cannot be solved
- Universal learning schemes are nonsense

However…

- Everyday experience
  - Universal learning seems to work (does it?)
- Conjecture: Property of physics
  - The universe seems biased

NFL-2: Living in a Non-Random World

Why Do We Need Priors?

Scenario "The Universe is indeed biased"

- We draw a function

$$y : \Omega(X) \to \{0,1\}$$

  from a small class

$$U \subset H_{all} \coloneqq \{y \mid y : \Omega(X) \to \{0,1\}\}$$

  where $\#U \ll \#H_{all}$.
- But we have no idea what U is.
  - So we consider all possible solutions $h \in H_{all}$
  - With uniform a priori likelihood

No Free Lunch (2)

Theorem ("no free lunch 2")

- Under these assumptions
  - "True" function sampled from a small set U
  - We have no knowledge about U (uniform prior on A)
- Averaged over all functions

$$H_{fit} = \{h \in H_{all} | \forall \mathbf{x} \in X_T : h(x) = y_i\}$$

  the expected generalization performance is

$$\frac{1}{\#H_{fit}} \sum_{h \in H_{fit}} L\big(h(\mathbf{x})\big) = 0.5$$

  (although the training error is zero)

Proof (NFL 2)

Consider subset that fits training data

$$H_{fit} = \{h \in A | \forall \mathbf{x} \in X_T : h(\mathbf{x}) = y(\mathbf{x})\}$$

- Consider a off-training point $\mathbf{x} \notin X_T$

  There are the same number of models $h \in H$ with

  $$h(\mathbf{x}) = 0 \quad \text{and} \quad h(\mathbf{x}) = 1$$

- Because of symmetry, just counting all fitting hs
  - For other $\mathbf{x}' \in X_T$: $h(\mathbf{x}') = y(\mathbf{x}')$ is fixed
  - For other $\mathbf{x}'' \notin X_T$: both $h(\mathbf{x}'') = 0$ and $h(\mathbf{x}'') = 1$ in H_{fit}
  - Overall: $\frac{1}{2}(\#\Omega(\mathrm{X}) - \#X_T)$ models the choice
- Thus, the average is 0.5
  - That is the case for every $\mathbf{x} \notin X_T$, which shows the claim

Summary NFL 1/2

Summary NFL

(1) No universal learning

- We cannot generalize a truly random labeling (ever)
  - No learning algorithm will be able to do this
  - No structure → no learning

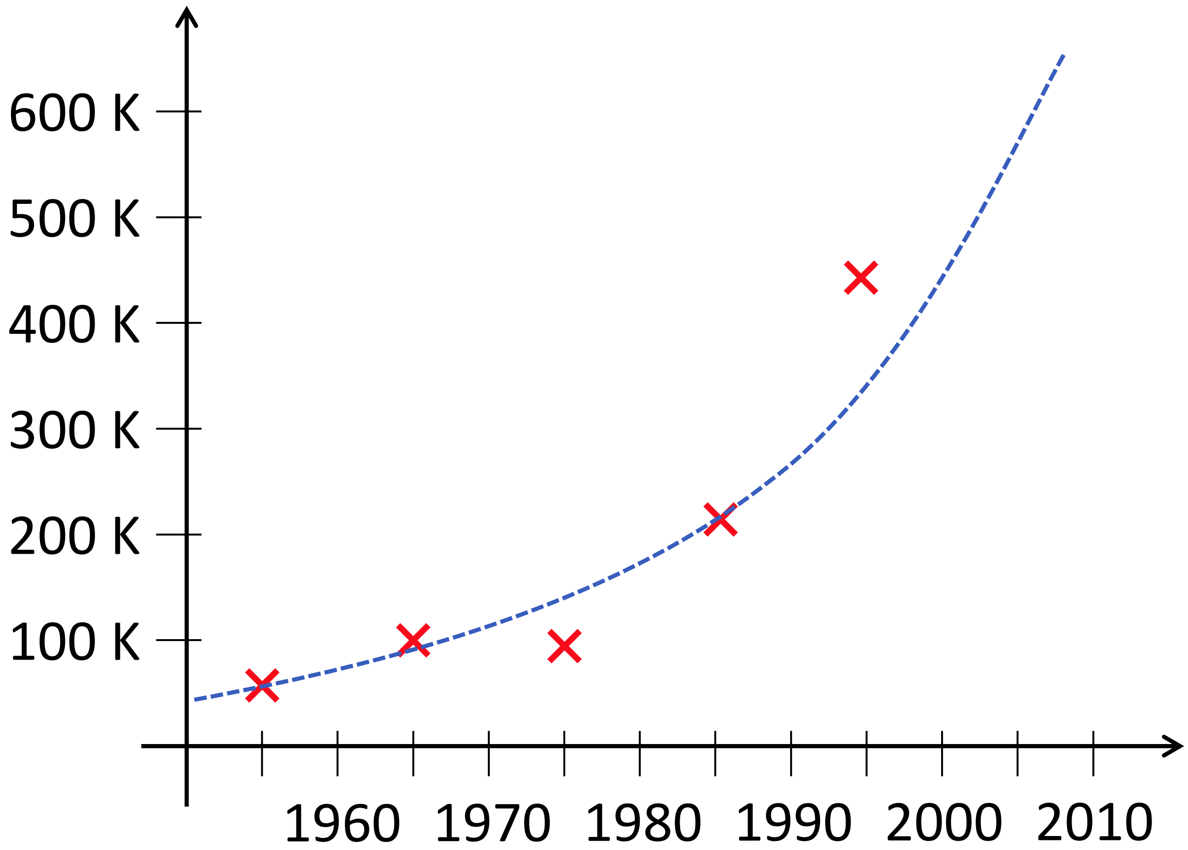
(2) No learning without priors

- We cannot generalize without prior assumptions
  - e.g.: probabilistic priors $P(h)$
  - e.g.: Model restrictions $h \in H, \quad \#H \ll \#H_{all}$
- Even if labeling drawn from a restrictive family
  - We need to know something about the structure
  - Will see soon: Gap ($\#H$ vs. $\#H_{all}$) is exponential in practice

Similar Arguments for other Settings

Example: Regression

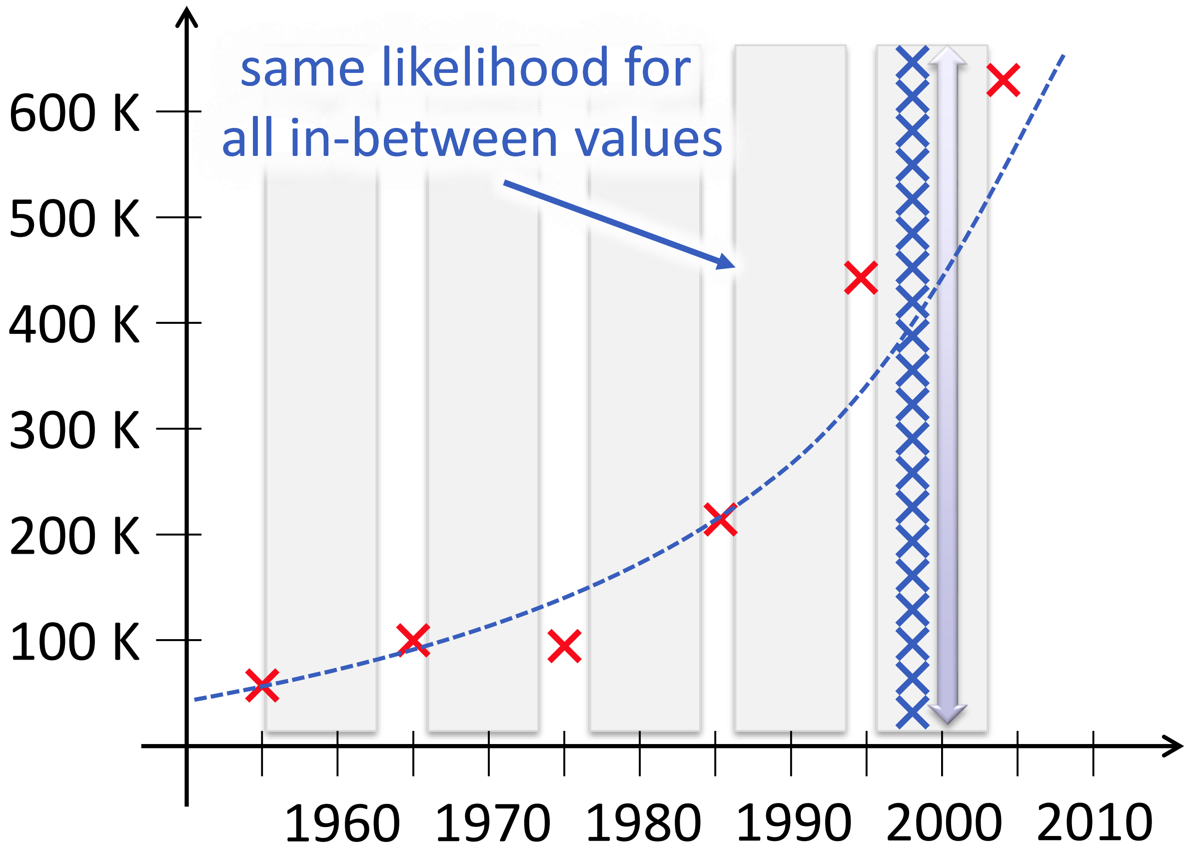
Housing Prices in Springfield *)



*) This is not investment advice

Example: Regression

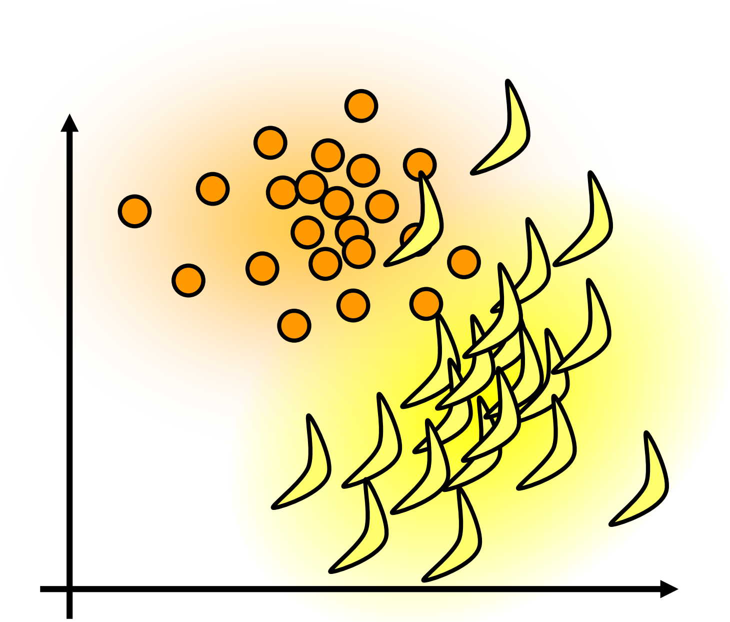
Housing Prices in Springfield *)



*) neither this

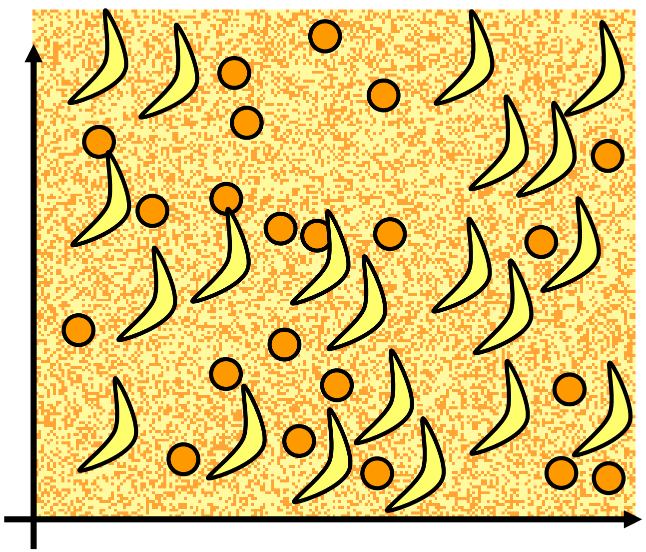
Example: Density Estimation (NFL-1)

Relativity of Orange-Banana Spaces



“smooth densities”
Here: Gaussians

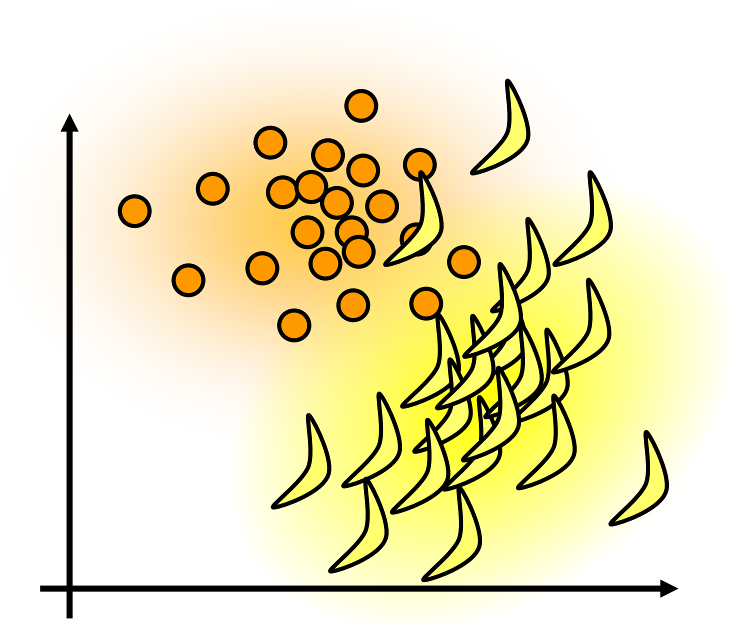
vs.



“random”
distributions

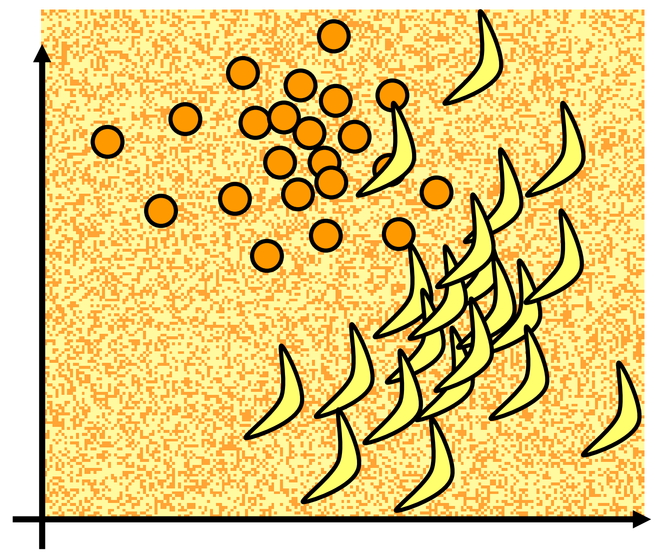
Example: Density Estimation (NFL-2)

Relativity of Orange-Banana Spaces



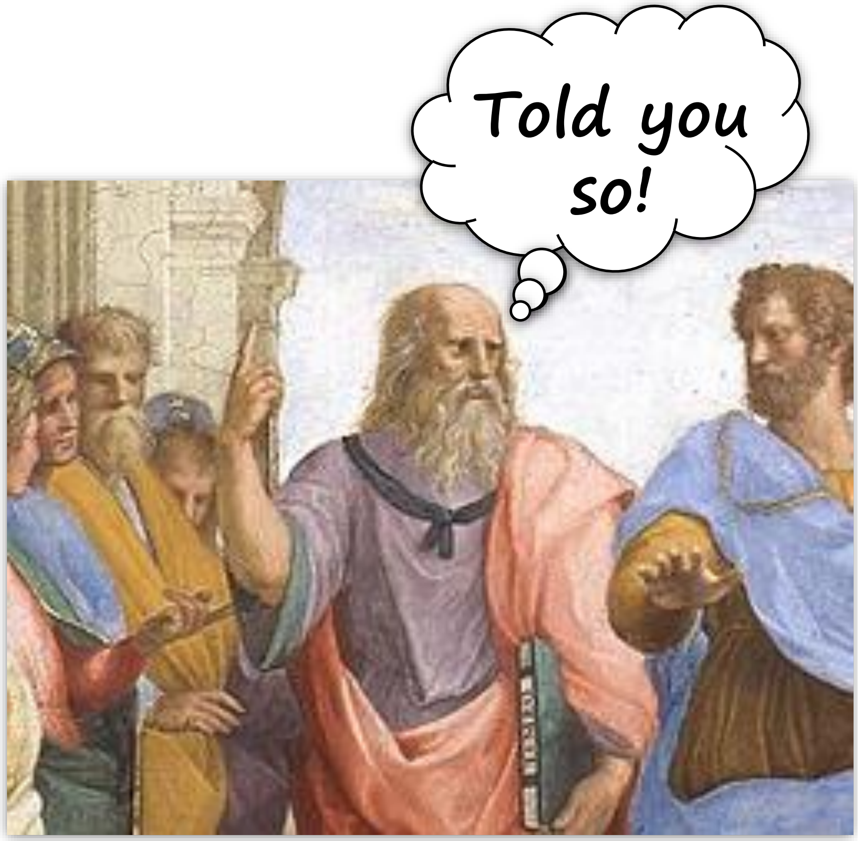
"smooth densities"
Here: Gaussians

vs.



"anything goes"
prior

The End.*)

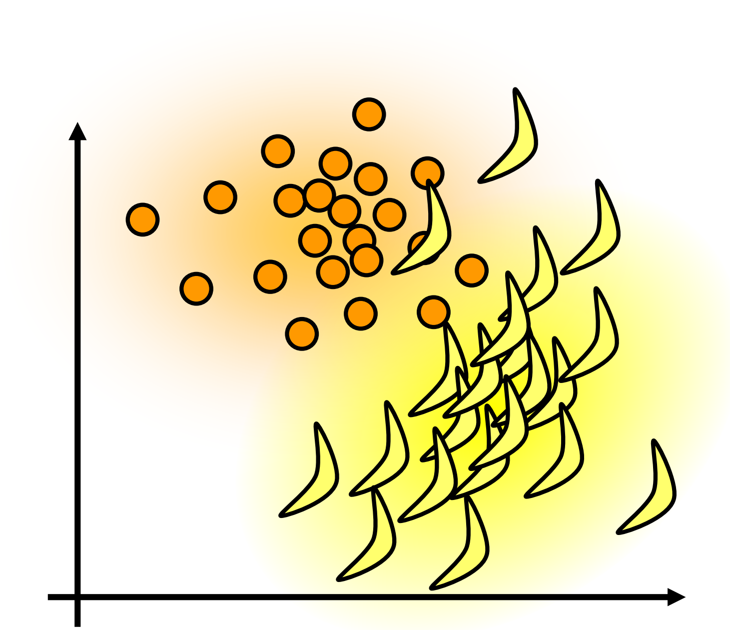


*) ML does not work.
Back to relational data bases!

Wait…

Example: Density Estimation

Say, we have observed the data (i.i.d.) below

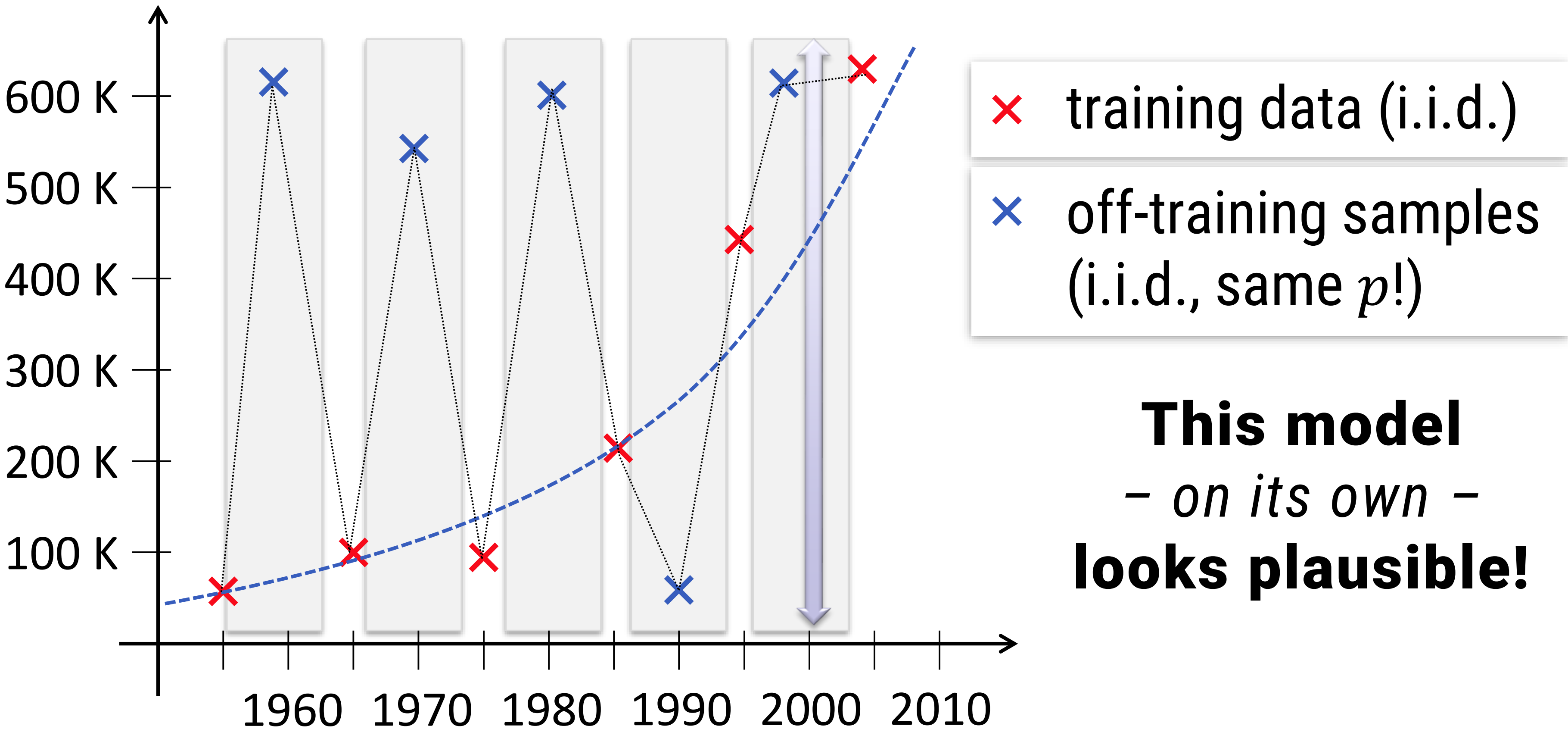


"smooth densities"
Here: Gaussians

This model
– on its own –
looks plausible!

Same here: Regression

Housing Prices in Springfield *)



This model
– on its own –
looks plausible!

*) neither this

Verification vs. Finding

We can objectively recognize good models

- Some oracle tells us one single model
- Performs consistently above chance on i.i.d. data

If this is true: Likely to generalize

- We know that it is likely to work on further i.i.d. data
- Can compute the odds for this holding in general

But: We cannot search for them universally

- If we consider all possible models, we cannot generalize

How many can we consider?

Summary

Conclusion

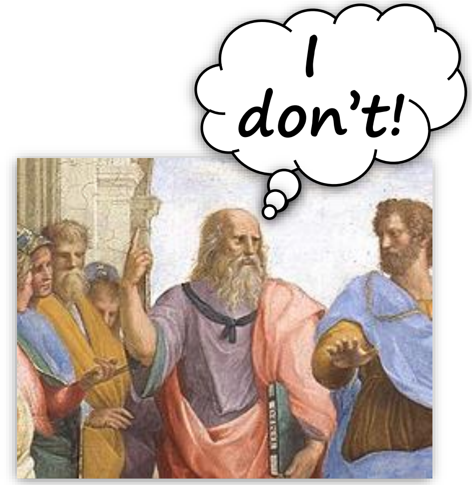
No Free Lunch

- No universal learning
  - Random problems cannot be solved
  - Unrestricted solutions will not work: Priors required
- Universal priors / learning schemes are nonsense

However

- We can quantify the likelihood of generalization
  - Depends on number of models considered during training
  - Next video: determine the odds
- Universal learning possible for restricted universes
  - Like ours: Human scientists believe in it

Conclusion



No Free Lunch

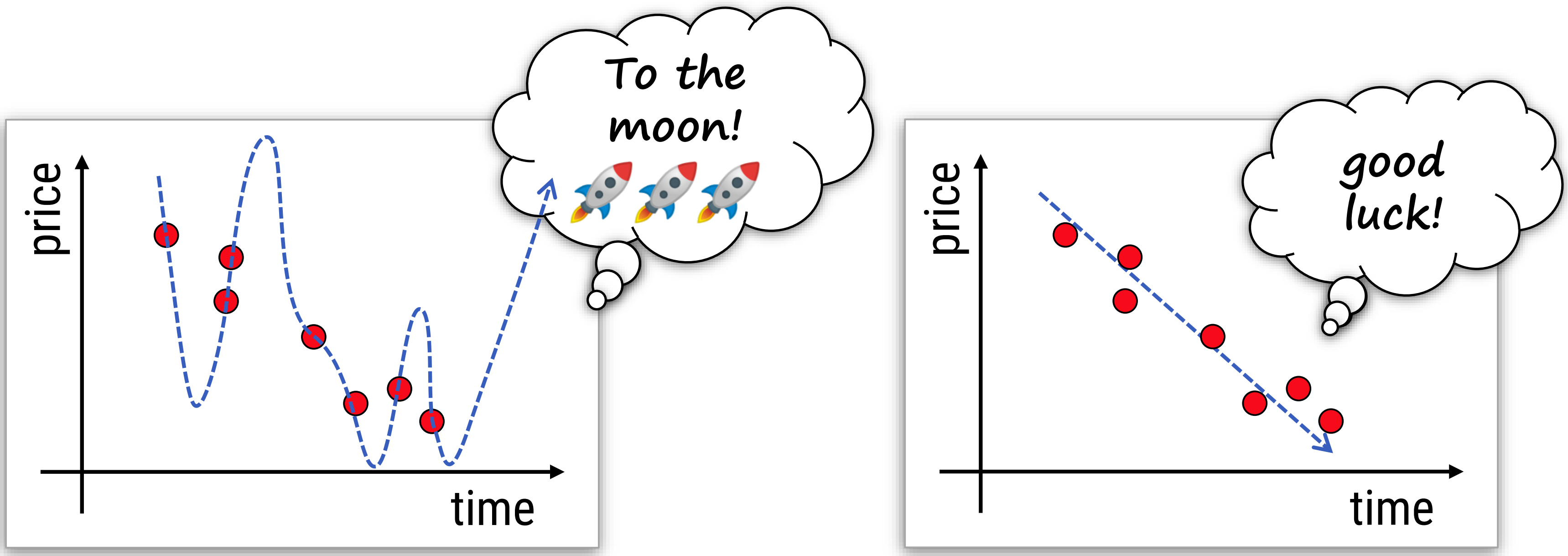
- No universal learning
  - Random problems cannot be solved
  - Unrestricted solutions will not work: Priors required
- Universal priors / learning schemes are nonsense

However

- We can quantify the likelihood of generalization
  - Depends on number of models considered during training
  - Next video: determine the odds
- Universal learning possible for restricted universes
  - Like ours: **Human scientists believe in it**

Modelling 2

STATISTICAL DATA MODELLING



Chapter 7

Generalization

Michael Wand · Institut für Informatik, JGU Mainz · michael.wand@uni-mainz.de

Video #07

Statistical Learning Theory

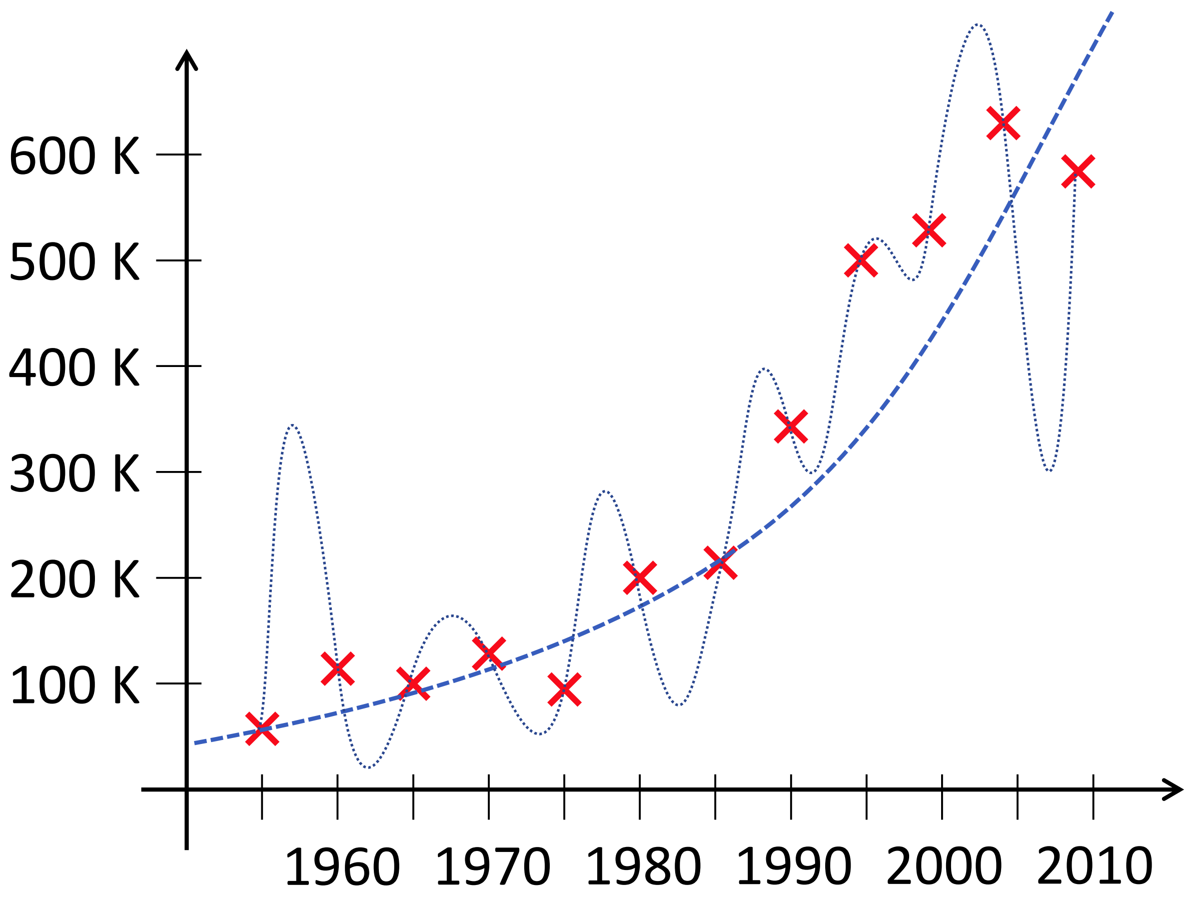
- **Limits:** No Free Lunch
- **Frequentist:** Statistical Learning Theory
- **Bayesian Model Selection**

Overfitting is Evil

…and to be avoided

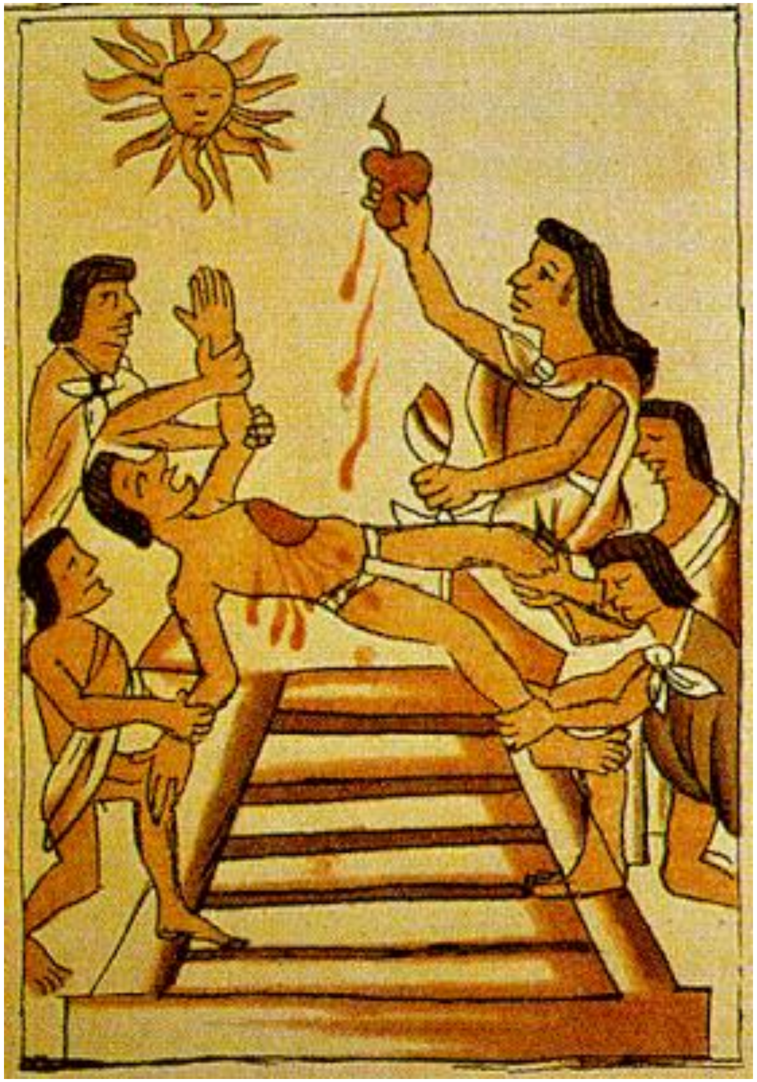
Regression Example

Housing Prices in Springfield



disclaimer: numbers are made up
this is not an investment advice

Overfitting



[source: https://commons.wikimedia.org/wiki/File:Aztecs10_sacrifice.gif]

Model Selection

How to choose the right model?

For example

- Linear, Quadratic, Higher order

We have seen

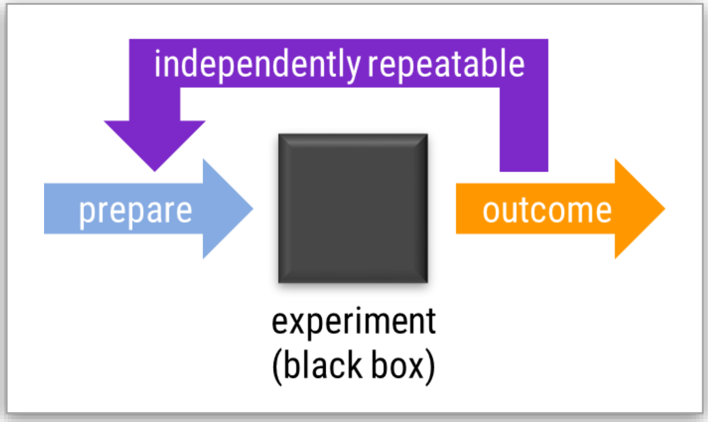
- Bayesian model averaging

Many other methods

- E.g.: cross validation (split in training/validation data)

But can we get an a priori guarantee?

SLT: Frequentist Bounds



Idea: Can't work every time by chance…

Answer: "Statistical Learning Theory"

- Objective bounds on generalization error
- Hence frequentist usage of statistics

SLT: Frequentist Bounds

“Probably Approximately Correct” (PAC)

- “PAC-learning” is a common model
- It tells us
  - That we will maintain a certain error ϵ
  - With certain likelihood δ
- Allows us to specify ϵ, δ
  - Tells us: minimum number of i.i.d. training examples n

SLT – Overview

Statistical Learning Theory

- Is a whole field of research
- This section gives only an introductory glimpse

Our goal

- To understand what is in principle possible
- And why
  - i.e., how to – roughly – prove that

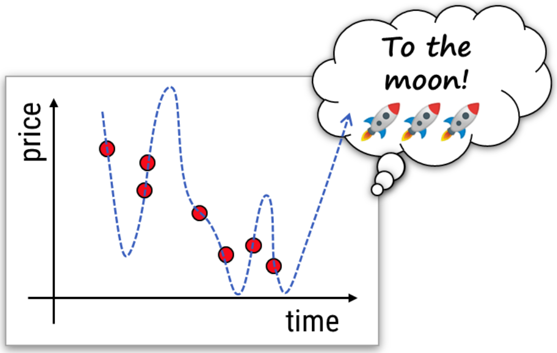
Bias-Variance Trade-Off

for General Regressors

Bias-Variance Trade-Off

Generalization error

- Training error might be misleading
- How reliable is the training error?



Bias-variance trade-off

Bias

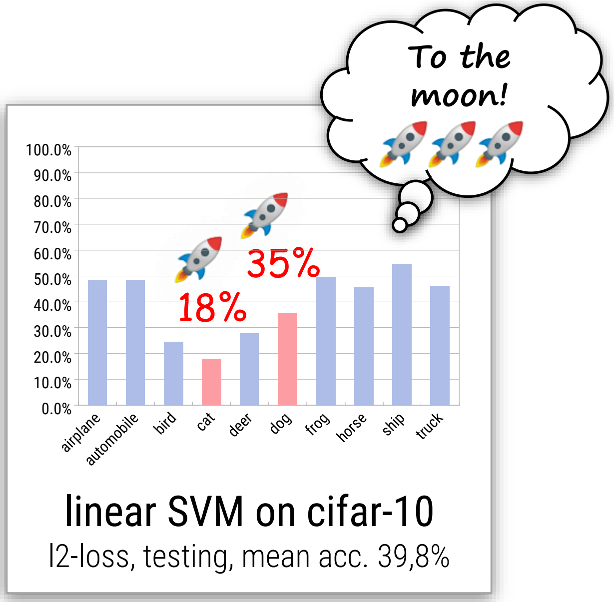
- Coarse prior assumptions to regularize model

Variance

- Bad generalization performance

Main Insight

*)



What is the problem

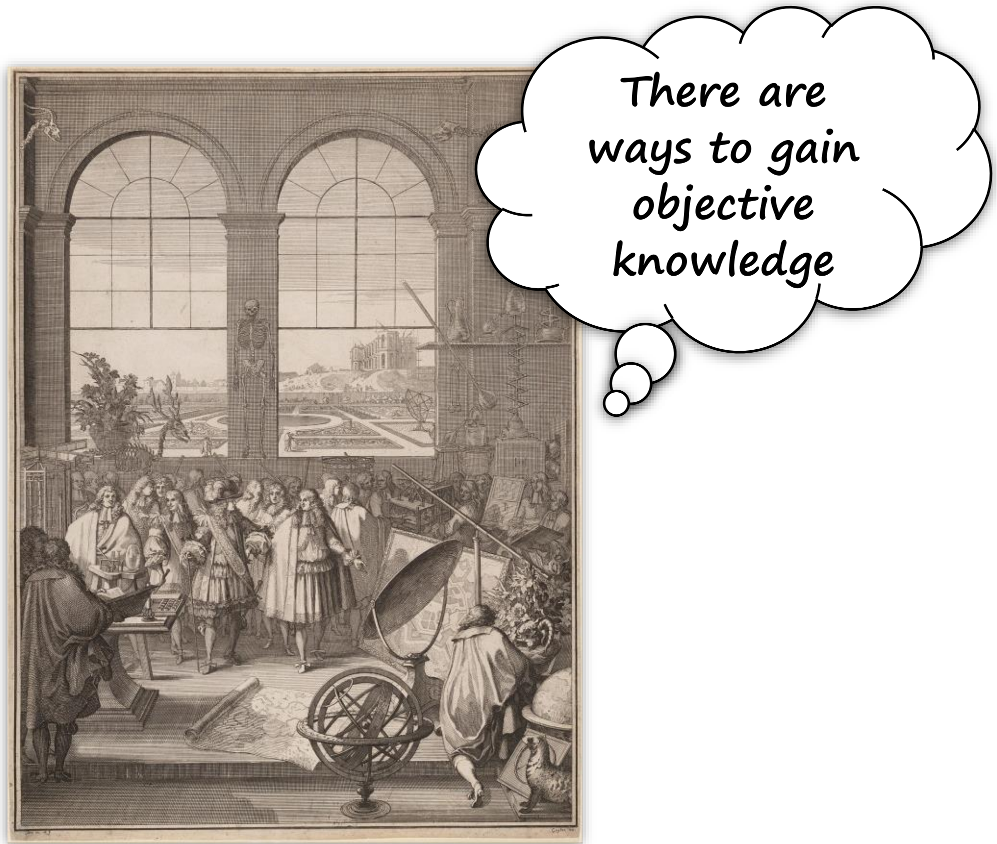
- Training error might be good "by chance"
- But generalization is still bad

Two sources of error

- We might not be able to find a good model
  - Try to fit a linear classifier to detect images of cats & dogs.
  - Good luck.*)
- We might not know the expected performance with sufficient precision

Wait…

We have seen that before!



Classical Statistics

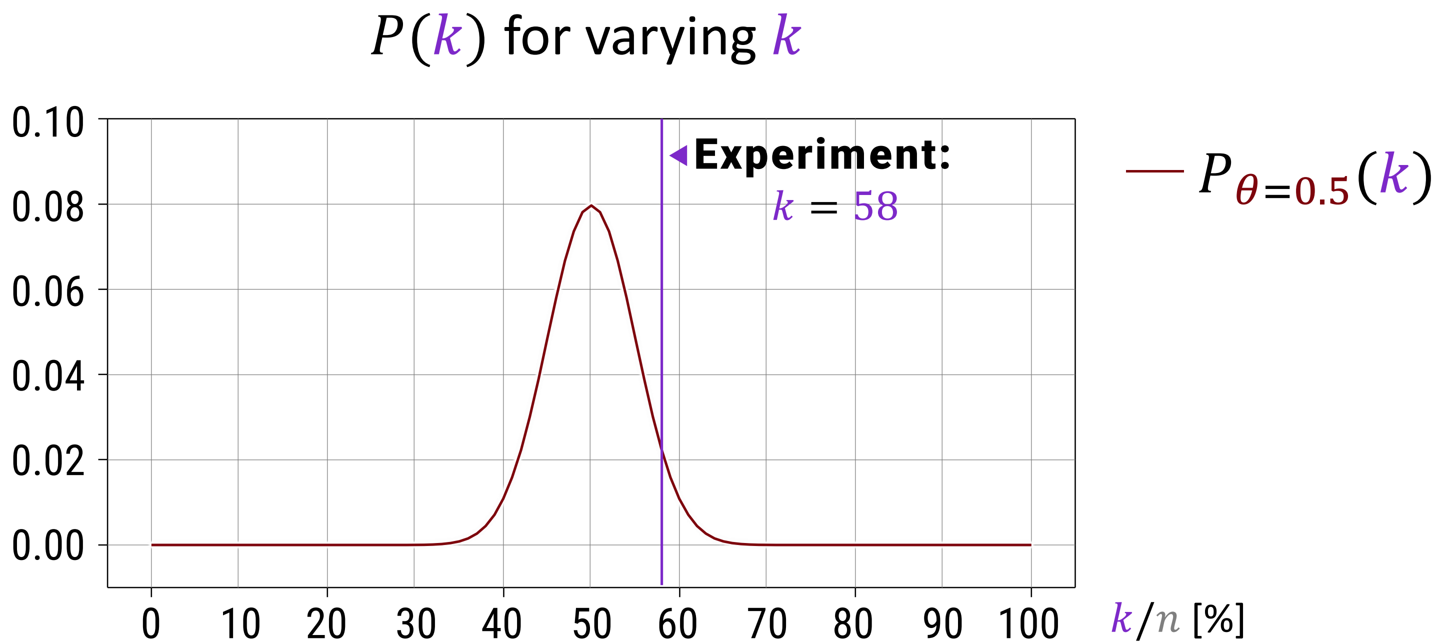
Two alternatives

- **Hypothesis:** The model performs at least this well
- **Null Hypothesis:** Just a random fluctuation

Frequentist Test

- Compute, how often we will see such fluctuations
- Shows how “significant” the observation was

Fair Coin Toss: What to expect



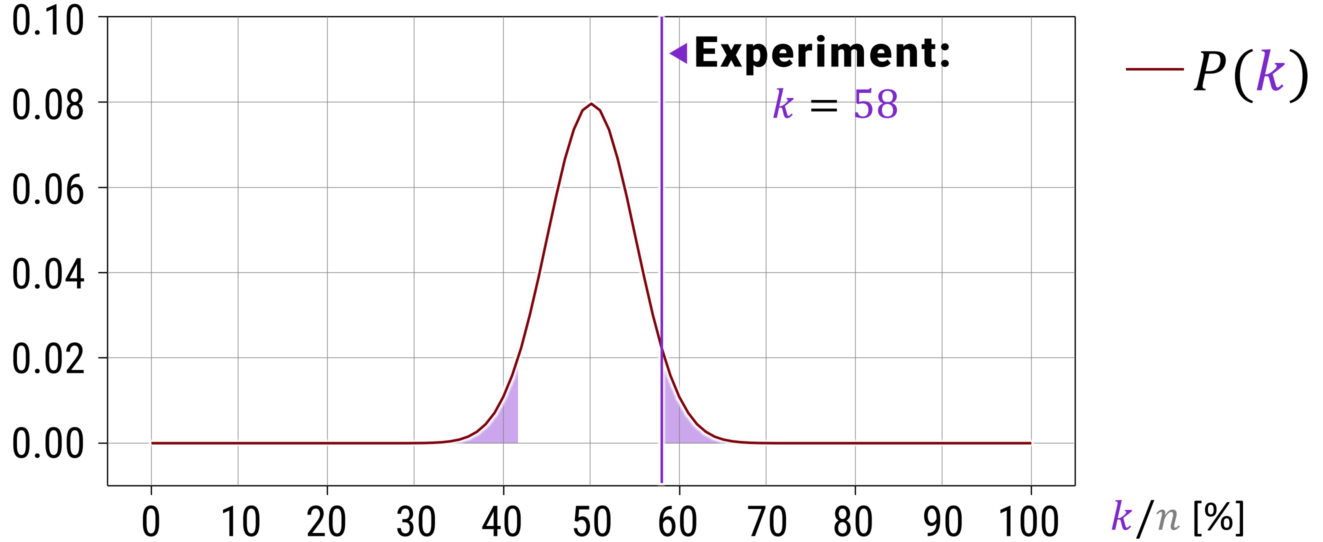
Baseline

- $n = 100$
- $\theta = 0.5$ (fair)

Experiment

- $n = 100$
- $k = 58$

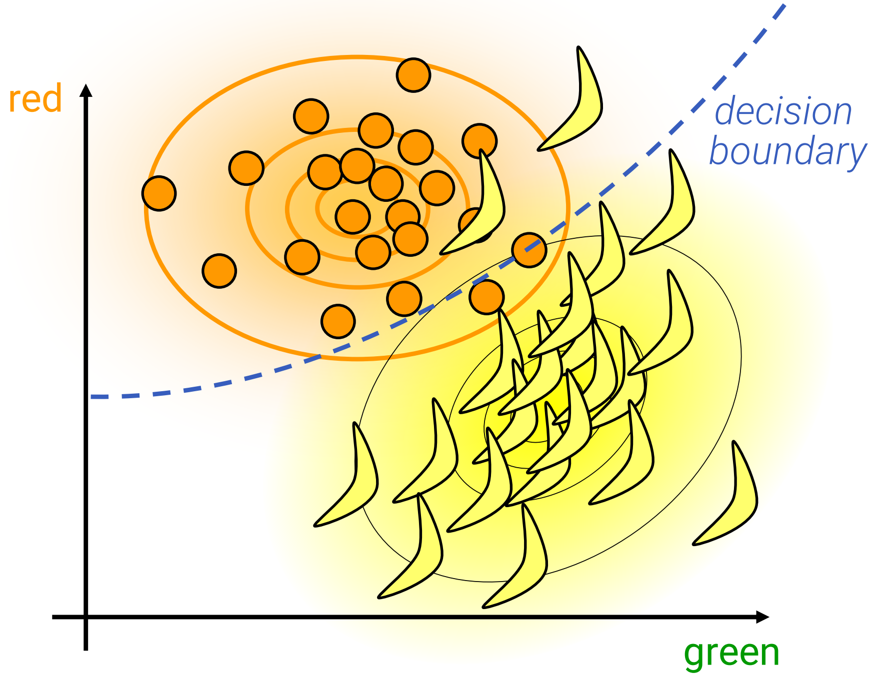
Two Sided Test



How often do we observe deviations $\Delta k \geq 8$?

$$P(|k-50| \geq K) = 2 \cdot \sum_{k=K}^{100} \binom{100}{k} \theta^k (1-\theta)^{n-k}$$

$$\approx 13\%$$



This Gaussian Model

Out of 40 i.i.d. fruit photos

- 20 banana, 20 oranges
- It classified 36 correctly
- It classified 4 wrongly

Likelihood for **pure chance?** → **Null-Hypothesis**

Binomial distribution: pick fruits, i.i.d., 50% banana

$$p(\text{"}\leq 4 \text{ wrong"}) = \sum_{k \in \{0,1,2,3,4\}} \binom{40}{k} 0.5^{k} 0.5^{40-k} = 9.3 \cdot 10^{-8}$$

What could possibly go wrong?

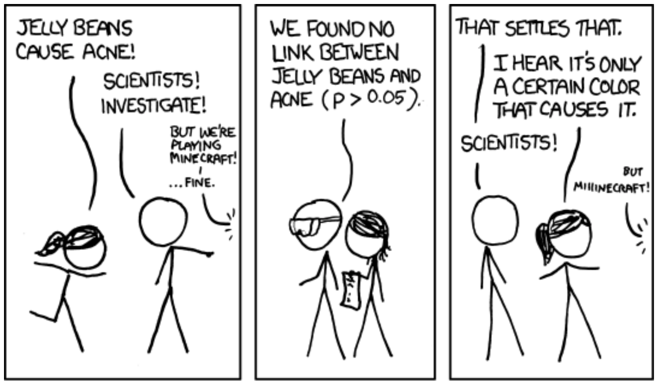
Does this solve our problem?

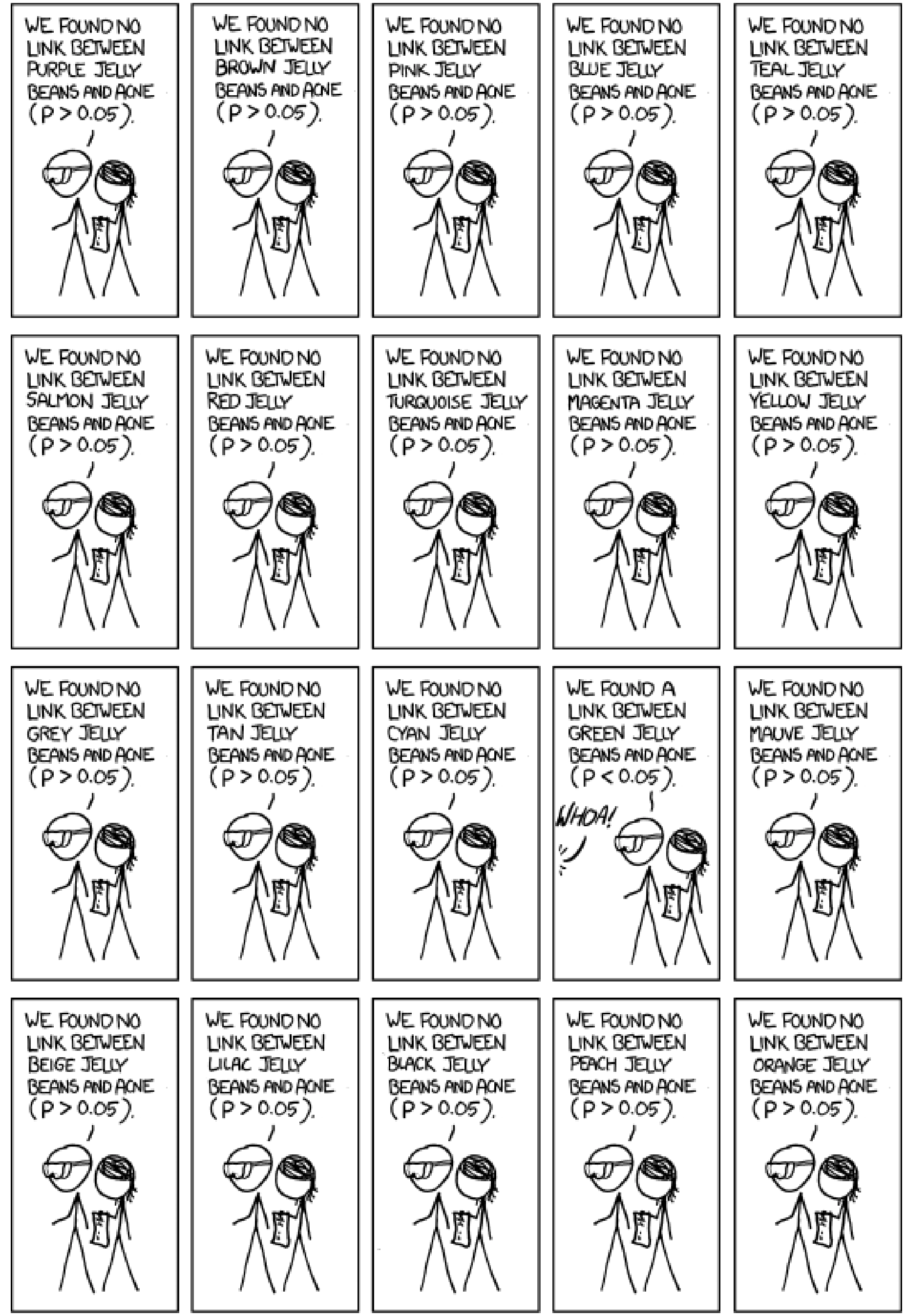
- No, because we want to fit a model
- We will choose from many models
- Evaluating only the best-performing one is not right

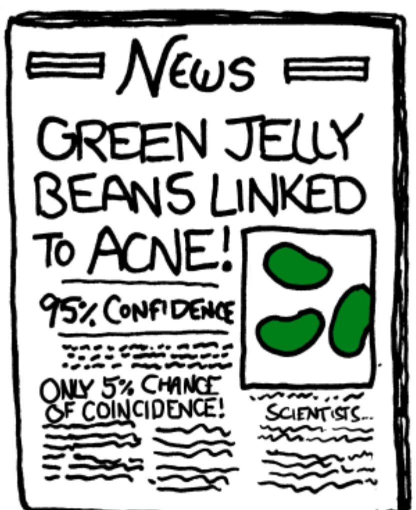
Illustrative: The extreme case

- We test all models
- Report only the best fitting
  - Which fits perfectly
- Obvious b.s. (bad science)

XKCD
“Green Jelly Beans”







Comic: [Randall Munroe, https://xkcd.com/882/]

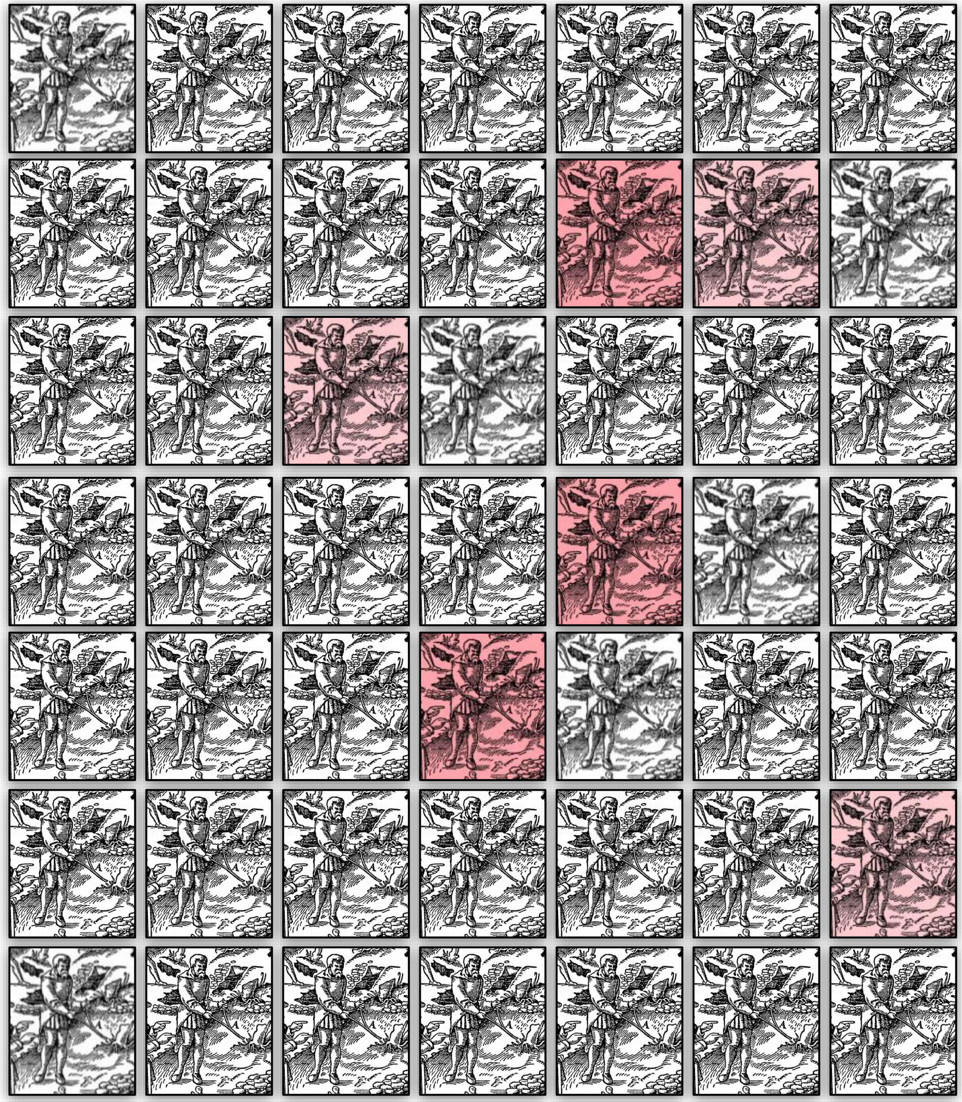
Speaking of Overfitting…

[Georgius Agricolas "De re metallica libri XII", 1556]

Multiple Hypothesis Testing

- Controversies are not uncommon
- Famous example: “Munich Dowsing Experiments”
  - https://en.wikipedia.org/wiki/Dowsing#Betz_1990_study

But those 6 guys



Multiple Hypothesis Testing

Machine Learning

- We have many potential models
- Formulations
  - Parameters $\theta \in \Omega(\theta)$
  - Or models $m \in M$
- Might even be continuous
  - $\theta \in \mathbb{R}^d$

How do we correct for this?

- Statistics: "Multiple Hypothesis Testing"
- Let's try this first…

Problem Formalization

Hypotheses & Losses

Learning Task

- Find function

$$f: \mathbb{R}^d \to \mathbb{R}^k$$

  from training data $(\mathbf{x}_1 \mapsto \mathbf{y}_1), \dots, (\mathbf{x}_n \mapsto \mathbf{y}_n)$

- Set of hypotheses

$$H \subset \{h \,|\, h: \mathbb{R}^d \to \mathbb{R}^k\}$$

- Loss functional

$$L: H \to \mathbb{R}, \qquad L(h) = \text{"how bad is } h\text{? "}$$

Hypotheses & Losses

Per data point

- Define loss $\ell(\tilde{\mathbf{y}}, \mathbf{y})$

$$\text{e.g.: } \ell(\tilde{\mathbf{y}}, \mathbf{y}) = |\tilde{\mathbf{y}} - \mathbf{y}|$$

Two types of losses

- Empirical loss

$$\hat{L}(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(h(\mathbf{x}_i), \mathbf{y}_i)$$

- Actual expected loss

$$L(h) = \mathbb{E}_{\mathbf{x}\sim p}\left[\ell\big(h(\mathbf{x}), f(\mathbf{x})\big)\right]$$

$$= \int_{\Omega} \ell\big(h(\mathbf{x}), f(\mathbf{x})\big) p(\mathbf{x}) d\mathbf{x}$$

$\ell(\tilde{\mathbf{y}}, \mathbf{y})$

point-wise loss

$\hat{L}(h)$

empirical loss

$L(h)$

expected loss

Hypotheses & Losses

$\hat{h}$
empirical risk minimizer

Best guesses

- Best guess (ERM)

empirical risk minimizer
(best according to what we know)

$$\hat{h} := \arg\min_{h \in \mathcal{H}} \hat{L}(h)$$

h^*
best possible hypothesis

- Actually best hypothesis

best fitting hypothesis available

$$h^* := \arg\min_{h \in \mathcal{H}} L(h)$$

Bias-Variance Trade-Off

Hypotheses & Losses

Best guesses

- Bias-Variance-Trade-Off

$$L(\hat{h}) = \underbrace{L(h^*)}_{\text{bias}} + \underbrace{L(\hat{h}) - L(h^*)}_{\text{excess loss}}$$

we do not have
any good model

we do not know how good
our models are

Hypotheses & Losses

Best guesses

- Bias-Variance-Trade-Off

$$L(\hat{h}) = \underbrace{L(h^*)}_{\text{bias}} + \underbrace{L(\hat{h}) - L(h^*)}_{\text{excess loss}}$$

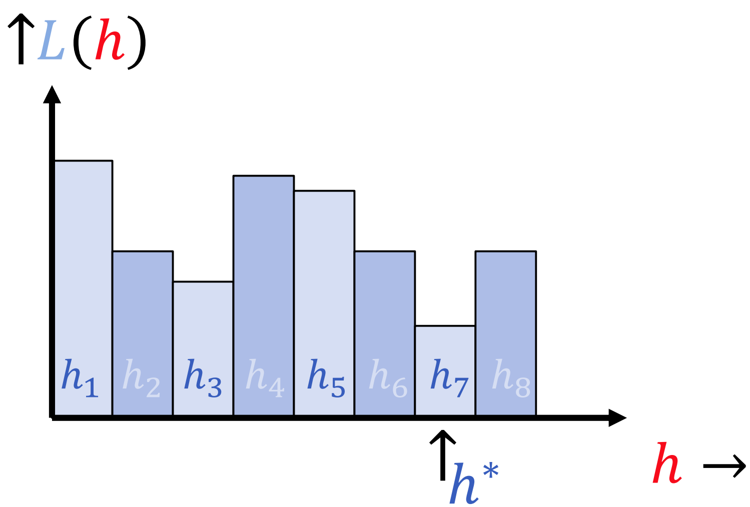
we do not have
any good model

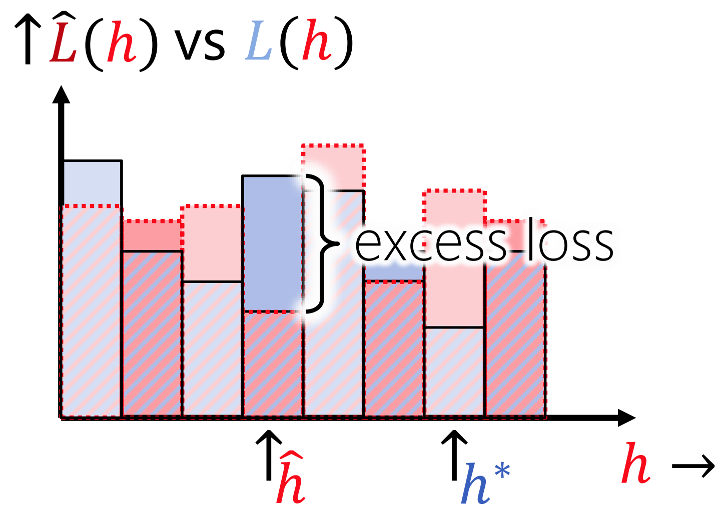
Know-how needed
for building good H

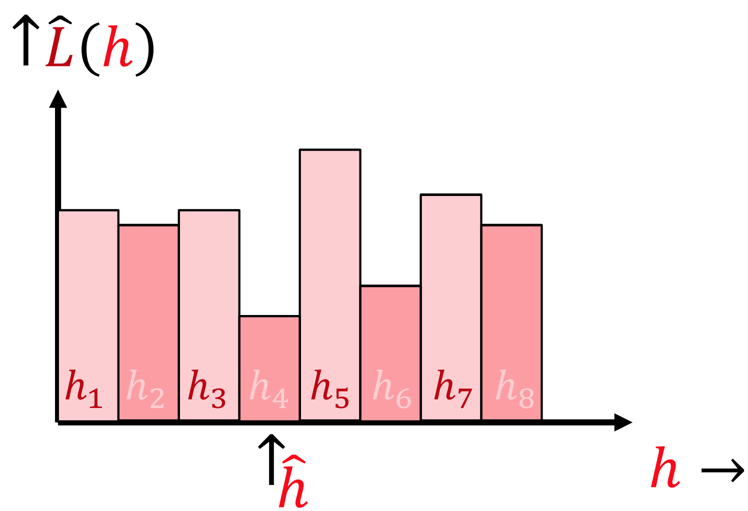
we do not know how good
our models are

Depends on training size n and
complexity of H ("overfitting")

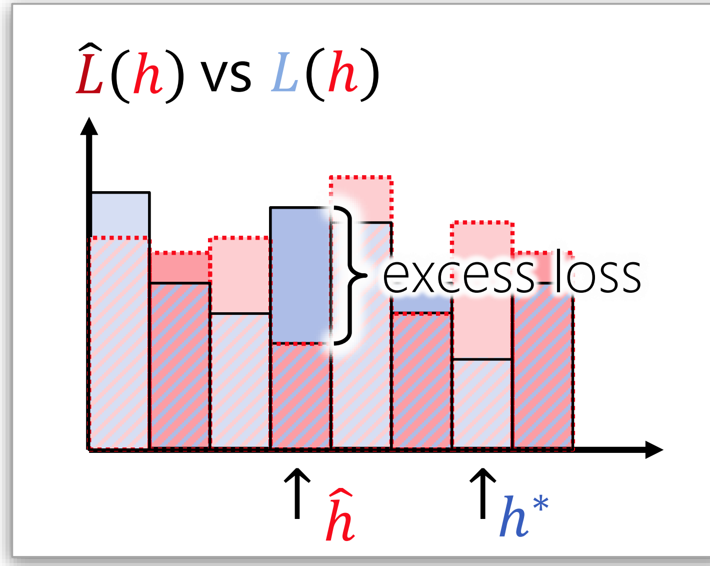
Culprit: Excess Loss







To Remember



Overfitting (in SLT terms)

- Excess loss too large
- Unable to pick good model
- Too much noise

What causes large excess loss?

- Too few data points
- Too many models in H

This is a trade-off!

Does this *require* noisy data?

- Observation of binary outcomes is already binomial!

Hypotheses & Losses

Theorem

- **Set H:** with $\#H$ hypothesis
- **Training data D:** n data points $\mathbf{x}_i, y_i = y(\mathbf{x}_i)$, i.i.d.
- **Bounded loss:** $\forall h, D: L(h) \in [0,1]$
- **Learn $h \in H$:** by empirical risk minimization

Then ⇒ excess loss bound

$$L(\hat{h}) - L(h^*) \le \sqrt{\frac{2\left(\ln(\#H) + \ln\frac{2}{\delta}\right)}{n}}$$

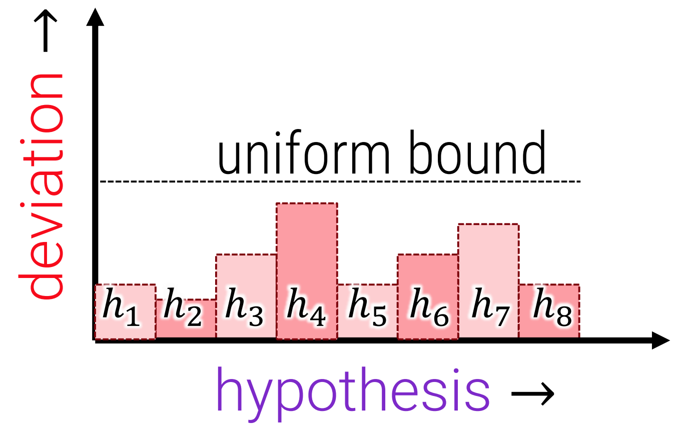
with probability $p \ge 1 - \delta$

Proof Sketch: "Uniform Error Bound"

Steps

- Empirical loss: $\hat{L}(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(h(x_i), y_i)$
- Asymptotically approx. normal distributed (CLT)
  - Expected error $\mathcal{O}(1/\sqrt{n})$
  - Deviation by factor c with prob. $\mathcal{O}(\exp(-c^2))$
- Multiple-hypothesis testing correction
  - Conservative assumption:
    P(any h_k overshoots) = $\sum_{k=1}^{\#H} P(h_k$ overshoots)
  - Union bound

Details



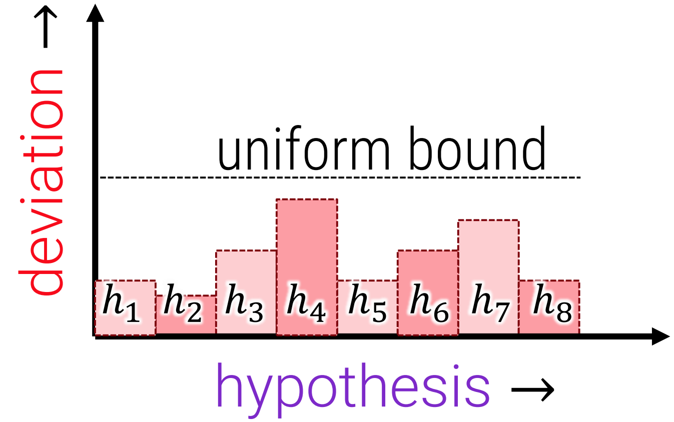
Single hypothesis

- Loss $\hat{L}(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(h(x_i), y_i)$
- Mean $L(h) = \mathbb{E}[\ell(h(x), y_i)]$
- Hoeffding inequality (think "CLT")

$$\text{with } \hat{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

$$\text{we get } P\left(\hat{X} - \mathbb{E}[\bar{X}] \geq \epsilon\right) \leq e^{-2n\epsilon^2}$$

Details



Single hypothesis

- Hoeffding inequality

$$P\left(\hat{X} - \mathbb{E}[\bar{X}] \geq \epsilon\right) \leq e^{-2n\epsilon^2}$$

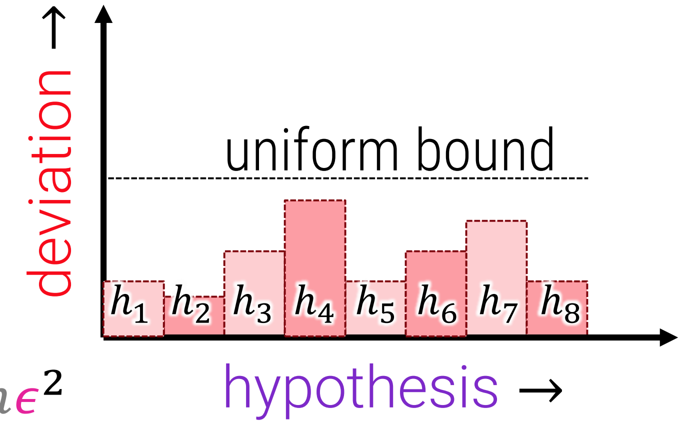
- Applied

$$P\left(\hat{L}(h_i) - L(h_i) \geq \epsilon\right) \leq e^{-2n\epsilon^2}$$

- Two sided

$$P\left(\left|\hat{L}(h_i) - L(h_i)\right| \geq \epsilon\right) \leq 2e^{-2n\epsilon^2}$$

Details



Multiple hypotheses

$$P\left(\left|\hat{L}(h) - L(h)\right| \leq \epsilon\right) \geq 1 - 2e^{-2n\epsilon^2}$$

- We now bound all $\#H$ hypotheses

$$P\left(\exists h \in H: \left|\hat{L}(h) - L(h)\right| \leq \epsilon\right)$$

$$\geq 1 - \sum_{h \in H} P\left(\left|\hat{L}(h) - L(h)\right| \geq \epsilon\right)$$

$$\geq 1 - (\#H) 2e^{-2n\epsilon^2}$$

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$
$$\leq P(A) + P(B)$$

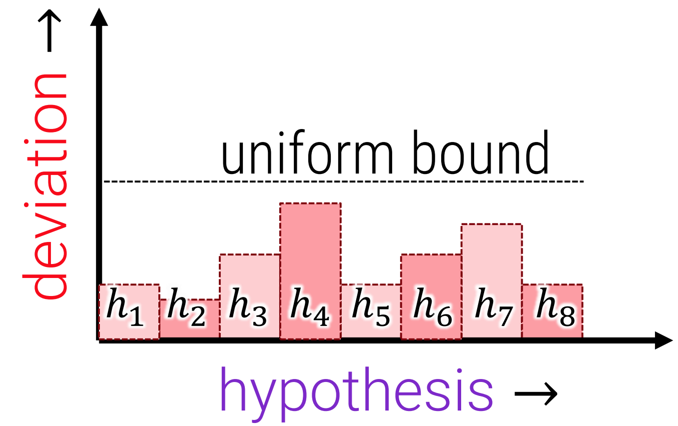
"union bound"

"every h is totally different"

Details

We got

$$P\left(\exists h \in H: \left|\hat{L}(h) - L(h)\right| \geq \epsilon\right)$$
$$\leq 1 - 2(\#H)e^{-2n\epsilon^2}$$



- Uniform error bound on all hypotheses

We use this…

- $$L(\hat{h}) - L(h^*) = L(\hat{h}) - \hat{L}(\hat{h}) + \hat{L}(\hat{h}) - \hat{L}(h^*) + \hat{L}(h^*) - L(h^*)$$

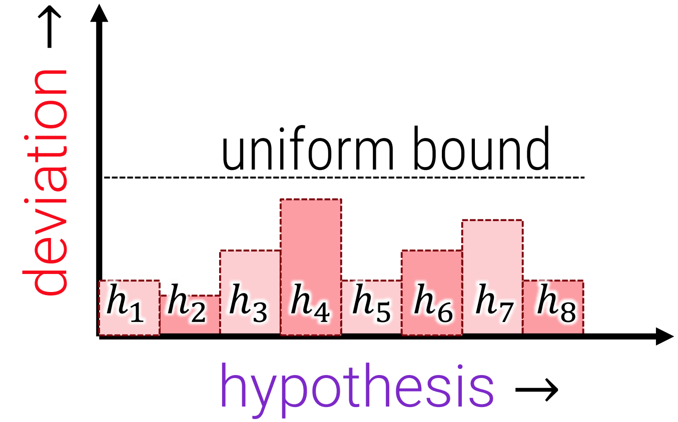
$= 0$

$= 0$

Details

We got

$$P\left(\exists h \in H: \left|\hat{L}(h) - L(h)\right| \geq \epsilon\right)$$
$$\leq 1 - 2(\#H)e^{-2n\epsilon^2}$$



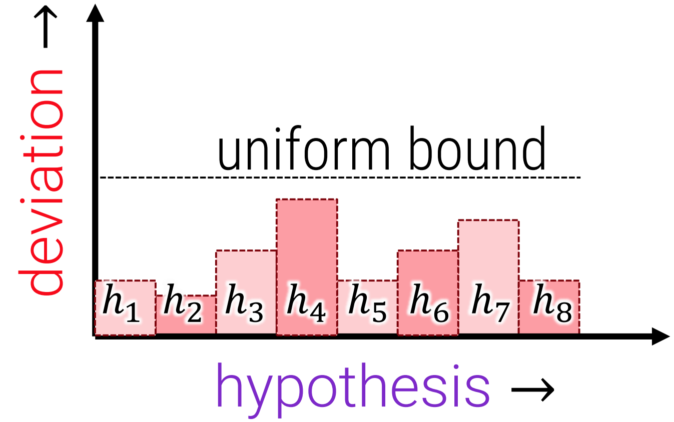
- Uniform error bound on all hypotheses

We use this…

- $$\begin{aligned} L(\hat{h}) - L(h^*) &= \left.L(\hat{h}) - \hat{L}(\hat{h})\right\} \leq \epsilon/2 \ ^{*)} \\ &+ \left.\hat{L}(\hat{h}) - \hat{L}(h^*)\right\} \leq 0 \ (\hat{h} \text{ is best wrt. } \hat{L}) \\ &+ \left.\hat{L}(h^*) - L(h^*)\right\} \leq \epsilon/2 \ ^{*)} \end{aligned}$$

*) we will choose ϵ later

Details



We got

$$P\left(\exists h \in H: \left|\hat{L}(h) - L(h)\right| \geq \epsilon\right)$$
$$\leq 1 - 2(\#H)e^{-2n\epsilon^2}$$

- Uniform error bound on all hypotheses

We use this…

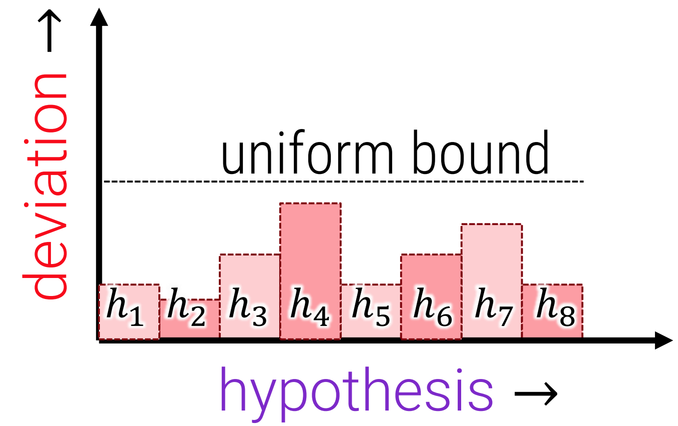
- $L(\hat{h}) - L(h^*) \leq \epsilon$ with probability $1 - 2(\#H)e^{-2n\cdot\frac{1}{4}\epsilon^2}$
- Bound should hold with probability $1 - \delta$

Details

We use this…

- $L(\hat{h}) - L(h^*) \leq \epsilon$

  with probability $1 - (\#H)e^{-\frac{1}{2}n\epsilon^2}$

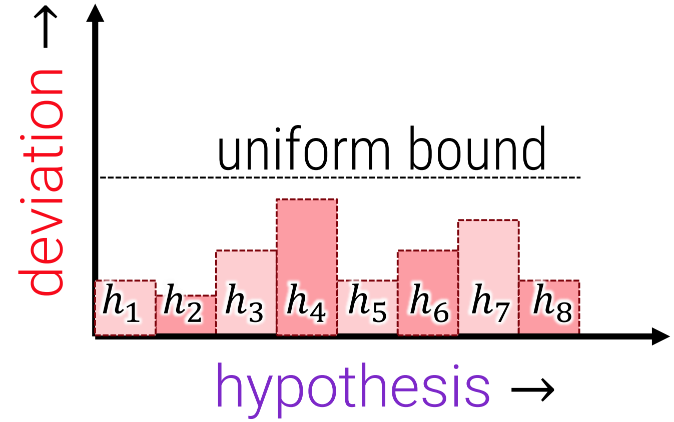


- Should hold with probability $1 - \delta$,

$$\text{i.e.,}\quad 2(\#H)e^{-\frac{1}{2}n\epsilon^2} \leq \delta$$

$$\text{i.e.,}\quad L(\hat{h}) - L(h^*) \leq \epsilon \leq \sqrt{\frac{2\left(\log(\#H) + \log\left(\frac{2}{\delta}\right)\right)}{n}}$$

Details



We use this…

- $L(\hat{h}) - L(h^*) \leq \epsilon$

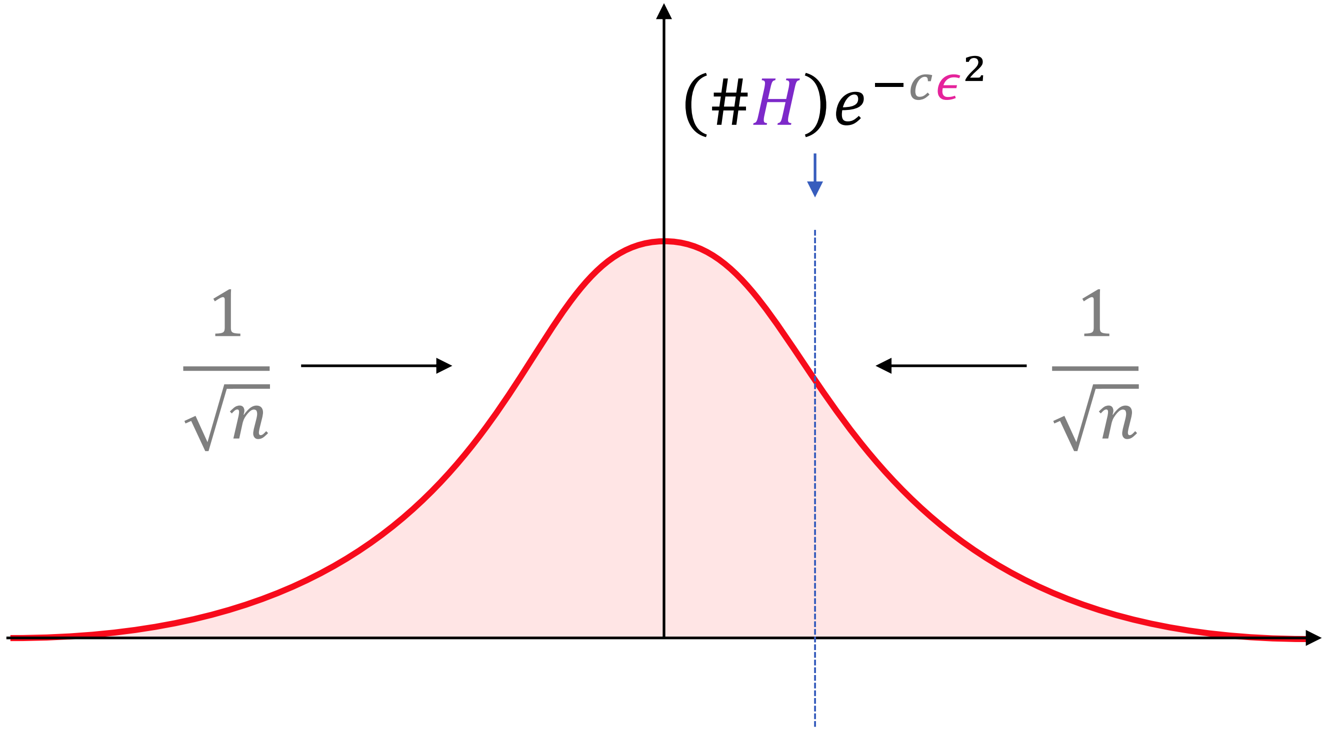
  with probability $1 - (\#H)e^{-\frac{1}{2}n\epsilon^2}$

- Should hold with probability $1 - \delta$

$$
\begin{aligned}
& 2(\#H)e^{-\frac{1}{2}n\epsilon^2} \leq \delta \\
\Rightarrow\ & -2(\#H)e^{-\frac{1}{2}n\epsilon^2} \leq -\delta \\
\Rightarrow\ & (\#H)e^{-\frac{1}{2}n\epsilon^2} \geq \frac{\delta}{2} \\
\Rightarrow\ & -\frac{1}{2}n\epsilon^2 \geq \log\frac{\delta}{2} - \log(\#H) \\
\Rightarrow\ & \frac{1}{2}n\epsilon^2 \leq \log(\#H) - \log\frac{\delta}{2}
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow\ & \frac{1}{2}n\epsilon^2 \leq \log(\#H) - \log\frac{\delta}{2} \\
\Rightarrow\ & \frac{1}{2}n\epsilon^2 \leq \log\left(\frac{2\#H}{\delta}\right) \\
\Rightarrow\ & \epsilon^2 \leq \frac{2}{n}\log\left(\frac{2\#H}{\delta}\right) \\
\Rightarrow\ & \epsilon \leq \sqrt{\frac{2\left(\log(\#H) + \log\left(\frac{2}{\delta}\right)\right)}{n}}
\end{aligned}
$$

Main Idea of the Proof



Consequences

Fixed errors ϵ, δ, determine n:

$$n \geq \frac{2}{\epsilon^2} \log\left(\frac{2\#H}{\delta}\right)$$

- We can compute a lower bound for the sample size
- Just solve inequality for n

Consequences

Bias-Variance Trade-Off

$$L(\hat{h}) \leq \underbrace{L(h^*)}_{\text{bias}} + \underbrace{\sqrt{\frac{2\left(\log(\#H) + \log\left(\frac{2}{\delta}\right)\right)}{n}}}_{\text{variance = excess loss}}$$

with probability $p \geq 1 - \delta$

- For loss functions $L \in [0,1]$
  - Other bounds: adapt analysis with rescaling
  - Unbounded, finite variance: Approximation via CLT
- Discrete set of hypotheses
- Bound might not be particularly tight

Consequences

Version for classification: Fits directly

- Finite set of $\#H$ models H
- Binary labeling problem: $y \in \{0,1\}$
- Use n i.i.d. data items (x_i, y_i) for training
- ERM: Choose model with lowest training error
- Trade-off for *generalization error* L

$$L(\hat{h}) \leq \underbrace{L(h^*)}_{\text{bias}} + \underbrace{\sqrt{\frac{2}{n}\ln\left(\frac{2\#H}{\delta}\right)}}_{\text{variance}} \quad \text{with } p \geq 1 - \delta$$

Continuous Models? (1)

Models classes are usually continuous

- $h = f_{\boldsymbol{\theta}}$ for $\boldsymbol{\theta} \in \mathbb{R}^d$

Simple argument: We are digital

- Each parameter θ_i is in $\mathbb{R} \approx$ `float32`.
  - 32 = O(1) bits
- Training set size $n \in \mathcal{O}(d)$ for d parameters
  - Exact numerical bound is rather loose anyways

Fancier argument

- ϵ-Covering of the function space

How about continuous models? (2)

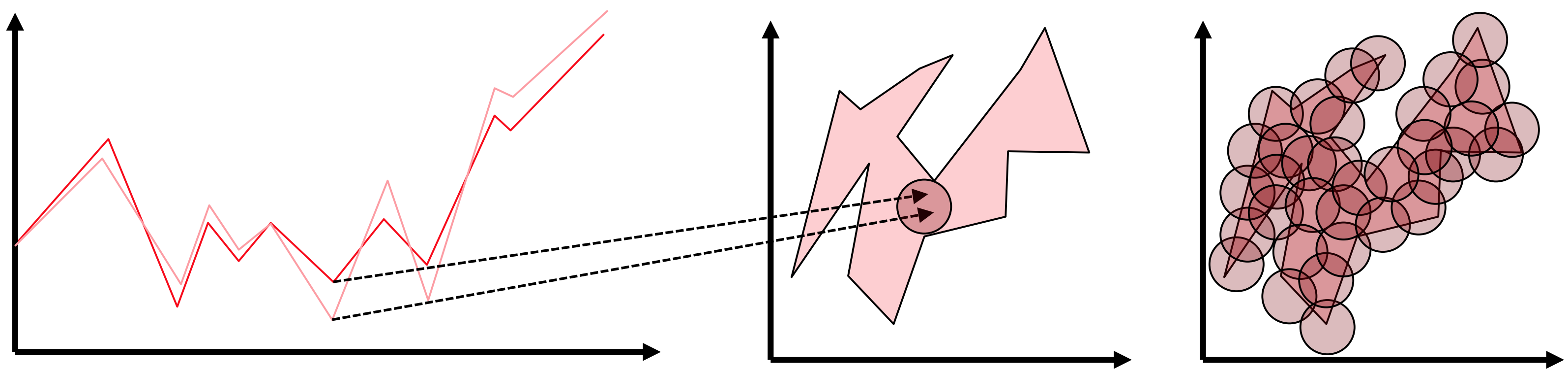
Very rough idea:

$$[h, (x, y)] \mapsto \ell(h(x), y)$$

- „Loss surface" varies with $h \in H$
  - We can have $h \in H = \{h_{\boldsymbol{\theta}} \big| \boldsymbol{\theta} \in \mathbb{R}^d\}$
  - But not every class H yields a useful bound
- Cover function space H with K ϵ-balls

$$B_\epsilon(h) = \left\{ h' \in H \middle| \begin{array}{c} \forall x \in \Omega(X), y \in \Omega(Y): \\ |\ell(h(x), y) - \ell(h'(x), y)| \leq \epsilon \end{array} \right\}$$

How about continuous models? (2)



Finite Covering of Function Space

- Assuming, we find a finite set

$$\mathrm{B} = \{B_\epsilon(h_1), \ldots, B_\epsilon(h_K)\} \text{ with } H \subseteq \cup_{i=1}^{K} B_\epsilon(h_i)$$

- We can substitute $\#H \leftarrow K$, but have additional error

$$L(\hat{h}) - L(h^*) \leq \mathcal{O}\left(\sqrt{\frac{2}{n}\ln\left(\frac{K}{\delta}\right)} + \epsilon\right) \text{ with } p \geq 1 - \delta$$

- We can search for best ϵ

Qualitative Analysis

Analysis

- Absolute numbers might not be tight

Qualitatively

$$\text{excess loss (variance)} \in \mathcal{O}\left(\sqrt{\frac{\log \# H}{n}}\right)$$

Two Theoretical Insights

Bias-Variance Trade-off

- Generalization error polynomial in model complexity:

$$j \text{ bits} \rightarrow K \leq 2^j \text{ models}$$

$$\rightarrow \mathcal{O}\left(\sqrt{\frac{1}{n}\log 2^j}\right) = \mathcal{O}\left(\sqrt{\frac{j}{n}}\right) \text{ error}$$

- Error $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ for n training examples

Relation to "No-Free-Lunch"

We get No-Free-Lunch back

- **Inputs** $\mathbf{x} \in \{0,1\}^d$ consists of d bits
  - 2^d different inputs possible
- **Labelings** $y: \{0,1, \dots, 2^d - 1\} \to \{0,1\}$
  - 2^{2^d} many classifications possible
  - Most flexible set of hypotheses: $\#H_{all} = 2^{2^d}$
- Can test K hypothesis with $n \in \mathcal{O}(\log K)$ samples$^{*)}$
- Asymptotics of no free-lunch:

$$K = 2^{2^d} \text{ (all possible: } H_{all}\text{)}$$

$$\to O(2^d) \text{ samples (all examples)}$$

$*)\ n \geq \frac{2}{\epsilon^2} \log\left(\frac{2\#H}{\delta}\right)$

More Complexity Analysis

This is also interesting

- Again, input $\mathbf{x} \in \{0,1\}^d$ (as "bitstring")
- Assume, we build a model that can fit

$$H_{all} = \{h_{\boldsymbol{\theta}} | \boldsymbol{\theta} \in \{0,1\}^M\}$$

  - Model is binary encoded in M bits
  - We need to encode 2^{2^d} models: need $M \geq 2^d$ bits
- Universal classifier will be infeasibly big

More Complexity Analysis

Considering H_{all} requires exponential data

- Lesson for $h \in H_{all}$:
  - $\mathrm{enc}(h) = 2^{\mathrm{enc}(\mathbf{x})}$
  - Training size $2^d = 2^{\mathrm{enc}(\mathbf{x})}$

  } exponential in input
- This also applies to a generative process!
  - Machine that generate
    - possible examples
    - and their labels
    - and can be tuned to any model, controlled by a bit-string
  - The description of this machine will also be exponential in $\mathrm{enc}(\mathbf{x})$

How Big Is the Gap?

Only polynomial-sized classifiers

- Have to shrink model size
- H from 2^{2^d} models to $\mathcal{O}\left(2^{\mathrm{poly}(d)}\right)$ models
  - Polynomial instead of exponential model size
  - Polynomial number of training examples
  - Realistically: linear
- Exponential gap
  - Prior knowledge must decrease $\#H$ from exponential in $\mathbf{enc}(\mathbf{x})$ to polynomial/linear
  - Uniform prior $P(X)$: Entropy from exponential to lin./poly.
- Exponentially more a priori knowledge than what we learn

Summary

Bias-Variance Trade-Off

To avoid overfitting

- Training set size n scales (worst-case) linearly with number of parameters (w/c. in bits)

To reduce randomness

- Increasing n reduces error by $\mathcal{O}\left(n^{-\frac{1}{2}}\right)$

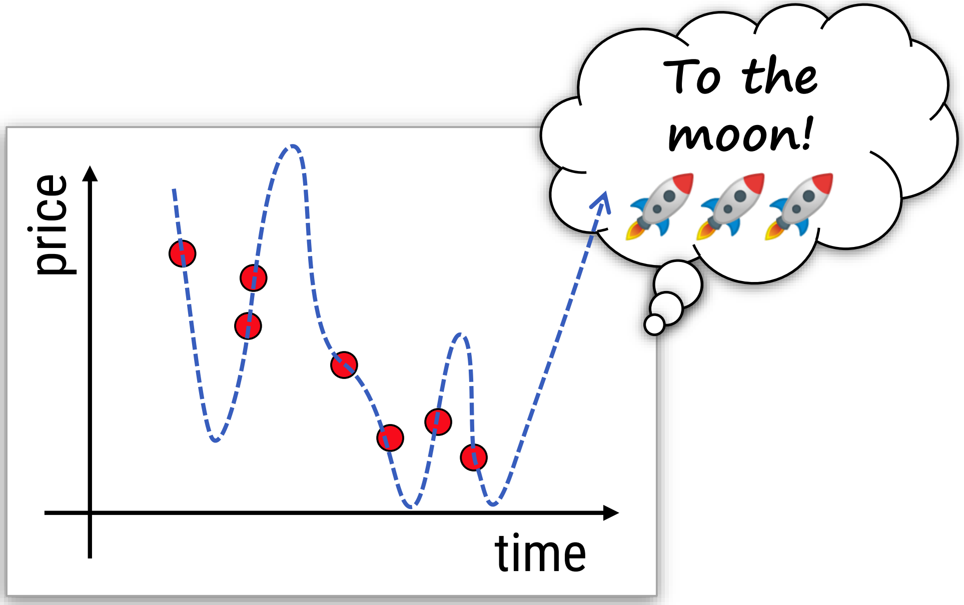
Prior knowledge

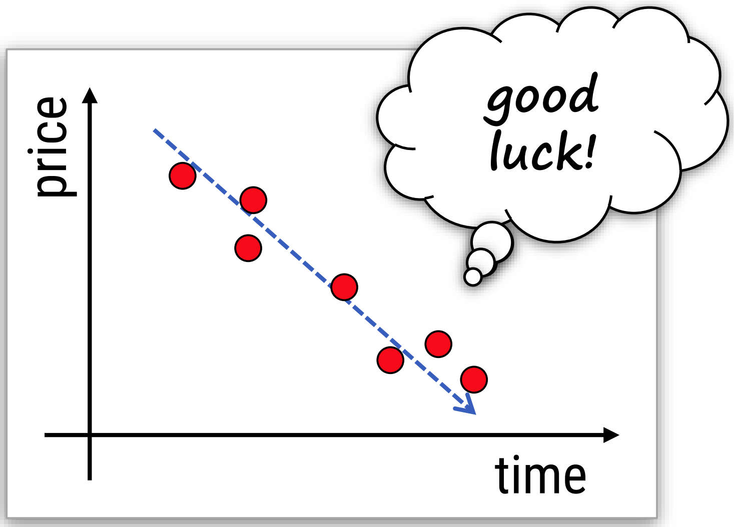
- To learn in realistic times, most of the knowledge must come from the prior

rule of thumb: $H\big(P(X)\big)$ linear in $\mathrm{enc}(\mathbf{x})$

Modelling 2

STATISTICAL DATA MODELLING





Chapter 7

Generalization

Michael Wand · Institut für Informatik, JGU Mainz · michael.wand@uni-mainz.de

Video #07

Statistical Learning Theory

- **Limits:** No Free Lunch
- **Frequentist:** Statistical Learning Theory
- **Bayesian Model Selection**

What We Have Learned So Far

No free lunch

- We cannot learn without (strong) priors

Generalization bounds

- Excess loss
  - Can prevent assessing generalization error
  - Bias-Variance-Trade-Off
- Sufficient: $\mathcal{O}(n)$ data points for model with n bits

Goal of this Section

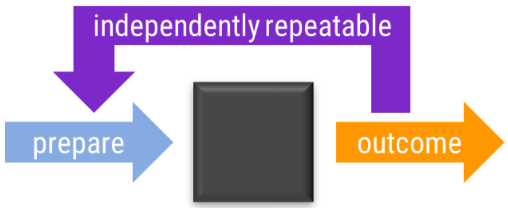
Understand better

- The bigger picture
  - Why are these bounds like this?
  - What is possible/impossible?
- How to select models
  - Adapt complexity automatically
- Bayesian model selection
  - How the Bayesian method works
  - What it can / cannot do for us
  - Information theoretical view
  - Looking back at the polynomial example

A Basic Information Theoretical View

“Frequentist” Model of Information

Experiment



experiment
(black box)

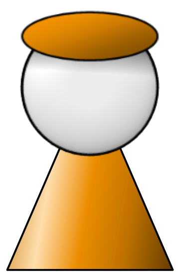
$\mathrm{enc}(x)$

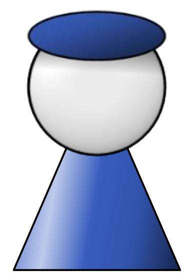
(channel)

Transmission

guys,
the outcomes are
x_7, x_{42}, x_{23}, x_8

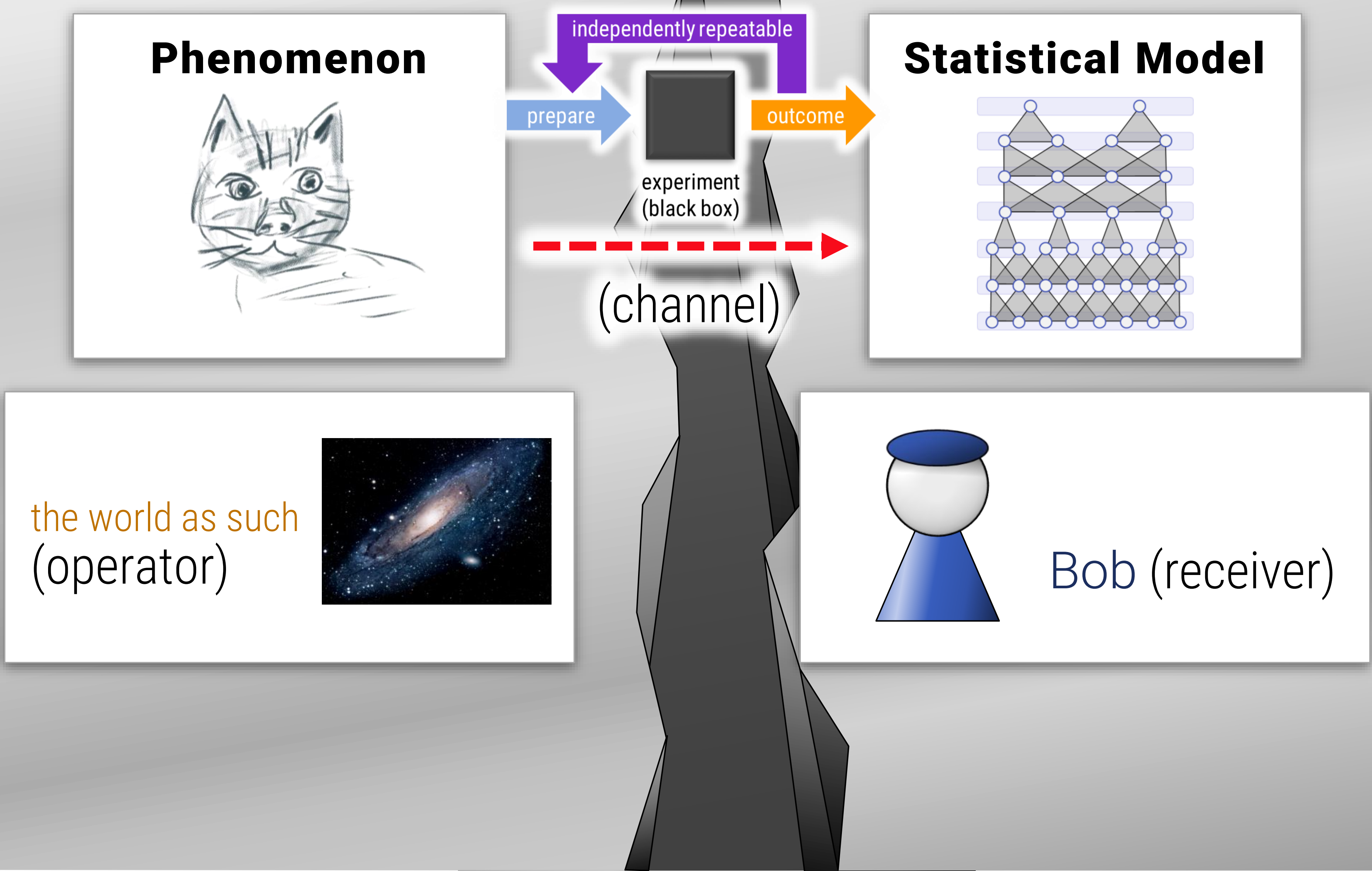
(operator) Alice



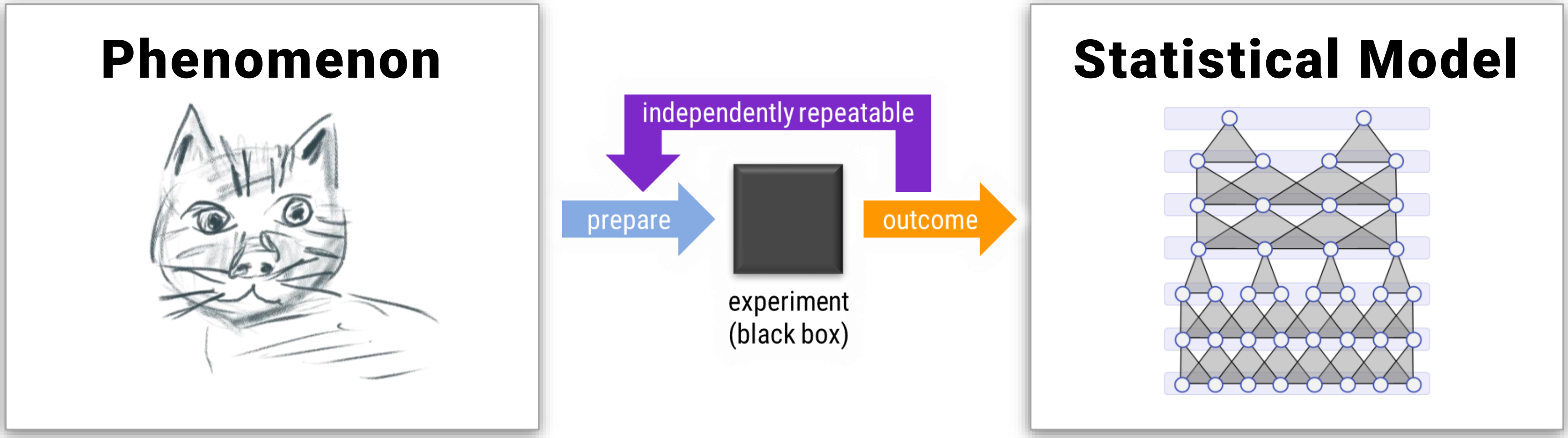


Bob (receiver)

The Experiment is the Channel



Back-of-the-Envelope Calculation



Information requirements

- Model has n bits of information (entropy)
- Need to draw n bits out of experiments

Back-of-the-Envelope Calculation

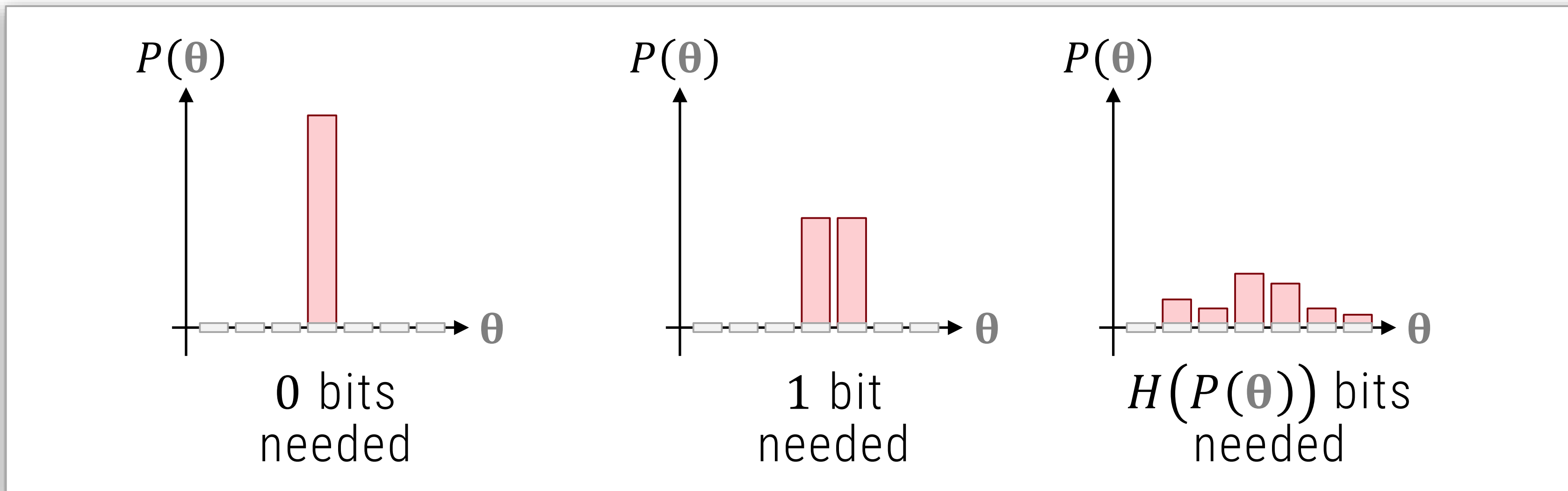
Information requirements

- Model has n bits of information
  - k equally likely hypotheses → $\log_2 k$ bits
  - Prior $p(\boldsymbol{\theta}) \rightarrow H(p) = \mathbb{E}_p[\log p]$ bits
    - Information that the prior cannot “fill-in”

Back-of-the-Envelope Calculation

Information requirements

- Model has n bits of information
  - k equally likely hypotheses → $\log_2 k$ bits
  - Prior $p(\boldsymbol{\theta}) \rightarrow H(p) = \mathbb{E}_p[\log p]$ bits
    - Information that the prior cannot "fill-in"



Back-of-the-Envelope Calculation

Information requirements

- Model has n bits of information
  - k equally likely hypotheses → $\log_2 k$ bits
  - Prior $p(\boldsymbol{\theta}) \to H(p) = \mathbb{E}_p[\log p]$ bits
    - Information that the prior cannot "fill-in"
- Need to draw n bits out of experiments
  - We get back at most $O(1)$ bits in every experiment
  - $\Omega(n)$ experiments necessary
  - $O(n)$ experiments sufficient to assess probability of successful predicting an output bit
- Conclusion: #data points ~ model entropy

Back-of-the-Envelope Calculation

Our goal now

- Build an automatic regularizer
- Ensure information criterion automatically
- Cannot break NFL: need prior model restriction

What Can Occam's Razor Do for us?

Occam's Razor

- "The simplest model fitting the data should be preferred"
- Keep models as simple as possible

Statistical Learning theory

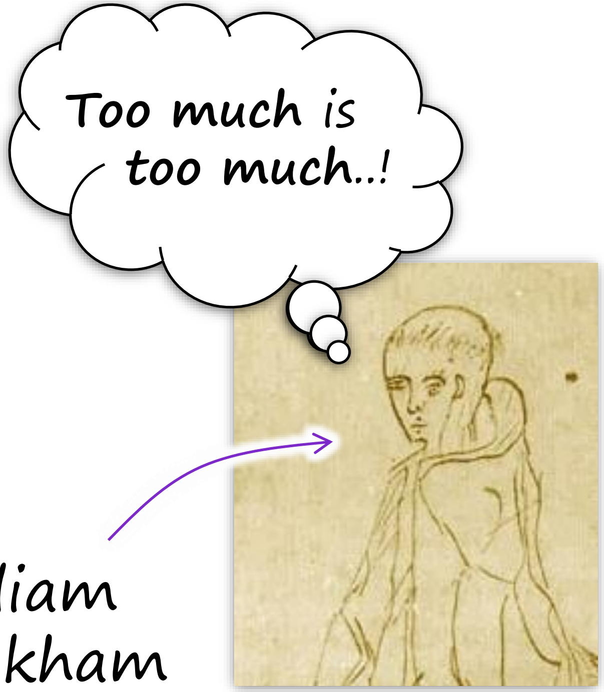
- Bounded complexity allows us to predict generalization performance
- It still might be very bad, but we know beforehand
- This is not a way to find models

What Can Occam's Razor Do for us?

Model Selection Scenario

- We have a *restricted* class of models
  - "All models" does not work – NFL-theorem!
- Within this class, models vary in complexity
  - Typically: assume that a "well-fitting" model is in this set
- We can automatically pick a suitable one
  - Complexity adapted to amount of data
  - Complexity adapted to difficulty of fitting
    - As simple as possible
- Results can be bad
  - Garbage (bad generalization), if set of models is unsuitable

MDL-Minimum Description Length



William of Ockham
(1287 – 1347)

[https://commons.wikimedia.org/wiki/File:William_of_Ockham_-_Logica_1341.jpg]

MDL Method

Minimum Description Length (MDL)

- Developed by Rissanen [1978]
- Try to keep models as simple as possible
- Simplified / tractable version of earlier ideas of Solomonov, Kolmogorov, Chaitin

Principle

- Encode data + model in the least amount of space
- Using entropy-coding as model (e.g. Huffman)

Literature:

Peter Grunwald: A tutorial introduction to the minimum description length principle. https://arxiv.org/pdf/math/0406077.pdf, 2004.

Solomonov Induction

Assumptions

- Data generated & recognized by algorithm
  - Universal Turing-machine (TM), incl. Python & C++
- Short models are best
  - Easiest to fit: preferred for statistical reasons
  - Easiest to find? “Universal” prior
- Bayes rule: Model M, Data D

$$P(M|D) \sim P(D|M)P(M)$$

$$P(M) = 2^{-|TM_{\min}(M)|}$$

$|TM_{\min}(M)|$ = Length of shortest TM computing M

Solomonov Induction

Properties

- Uncomputable
  - $|TM_{\min}(M)|$ cannot be computed
- Asymptotically invariant
  - Length of TM only vary by additive constant
  - Simulator for TM in a universal TM needs O(1) space
- “Radical” formalization of Occam’s Razor

Variants

- AIXI – Reinforcement learning (M. Hutter)
- Speed-Prior: short-running TMs first (J. Schmidhuber)
  - Exponential instead of impossible

Rissanen's MDL

Minimum Description Length

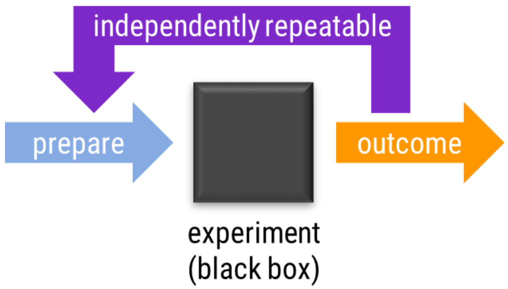
- Probabilistic measurement of data d
  - n i.i.d. repeats
- Looking for best model m
  - m needs parameters $\boldsymbol{\theta}$
  - Shortest message describing all n experiments
  - Optimal choice of m depends on n

Practical method

- No inherent computability issues
- Machine model implicit in "coding unit"

Back to the Standard Model…

Experiment



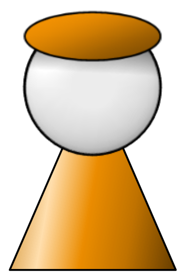
enc(**d**)

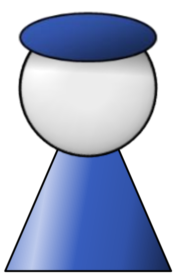
(channel)

Transmission

guys,
the outcomes are
x_7, x_{42}, x_{23}, x_8

(operator) Alice





Bob (receiver)

Formalization

Experimental Setup

- Model m out of set $M = \{m_1, \dots, m_k\}$
- Each model has parameters $\boldsymbol{\theta}$
- Data $\mathbf{d} = (d_1, \dots, d_n)$

Minimum Description Length Method

- Alice sends outcome $\mathbf{d}$ to Bob using model m
  - Send model m
  - Send model parameters $\boldsymbol{\theta}$
  - Send data $\mathbf{d}$
    - Using the model: "residuals" to model mean
    - Probabilistic codes for $P(d_i | m, \boldsymbol{\theta})$

Information Theory

Reminder

- Probability distribution $p(x)$
- Information $I(x) = -\log p(x)$
- Expected information = Entropy

$$H(p) = -\sum_{x \in \Omega(X)} p(x) \log p(x)$$

Coding Theorem

- Can encode outcomes x with expected $\lceil H(p) \rceil$ bits
- Constructive proof: Huffman coding

MDL Formalization

How to send?

- There are k models.
  - Need at most $\lceil \log_2 k \rceil$ bits
- Parameters $\boldsymbol{\theta}$ for model m have N_m bits: $\boldsymbol{\theta} \in \{0,1\}^{N_m}$
  - At most $\lceil \log_2 N_m \rceil$ bits
  - N_m depends on / describes model complexity!
- Observations have N bits: $d_i \in \{0,1\}^N$
  - At most $n\lceil \log_2 N \rceil$ bits
  - But we can do better, and this is important!

MDL Formalization

$$\underbrace{enc(m)}_{\text{fixed}}\ \underbrace{enc(\boldsymbol{\theta})}_{\text{model complexity}}\ \underbrace{enc(d_1)\cdots enc(d_n)}_{\text{data fit}}$$

Sending a model

- Encode choice of m
  - Binary number with $\lceil \log_2 k \rceil$ bits
  - Message length $L(m) = \lceil \log_2 k \rceil$
- Encode parameters $\boldsymbol{\theta}$
  - Using $\lceil \log_2 N_m \rceil$ bits
  - Large N_m means larger messages
  - Message length $L(\boldsymbol{\theta}|m) \leq \lceil \log_2 N_m \rceil$
- Encode data $\mathbf{d}$
  - Using distribution $P(\mathbf{d}|m, \boldsymbol{\theta})$
  - Using $L(\mathbf{d}|m, \boldsymbol{\theta}) = H\big(P(\mathbf{d}|m, \boldsymbol{\theta})\big)$ bits

Model Selection

$$\underbrace{enc(m)}_{\text{fixed}}\;\underbrace{enc(\boldsymbol{\theta})}_{\text{model complexity}}\;\underbrace{enc(d_1)\cdots enc(d_n)}_{\text{data fit}}$$

Wo do not ever send models

- This is just a thought experiment
- We choose model m such that message length

$$L(m) + L(\boldsymbol{\theta}|m) + L(\mathbf{d}|\boldsymbol{\theta}, m)$$

  is minimized

Analysis

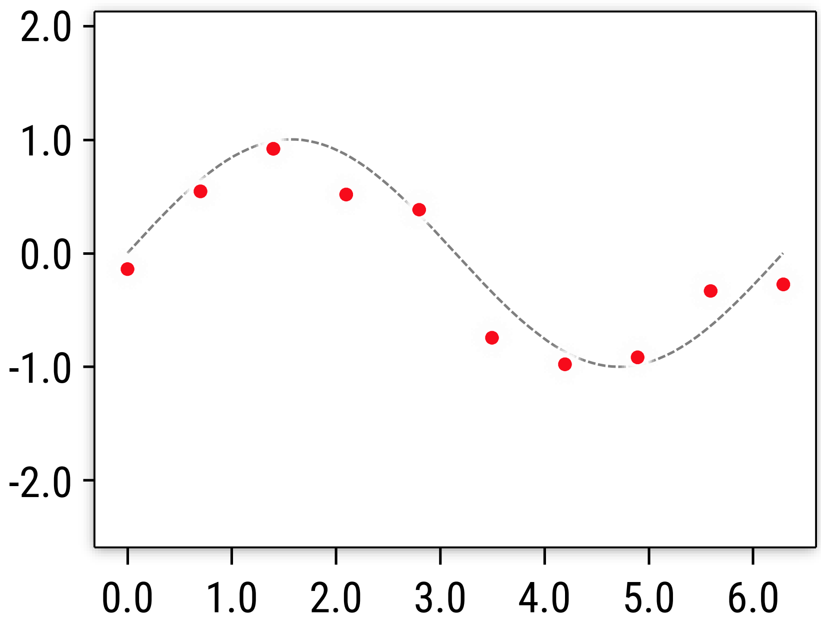
$$\underbrace{L(m)}_{\substack{\text{constant}\\\text{(for now)}}} + \underbrace{L(\boldsymbol{\theta}|m)}_{\substack{\text{grows with}\\\text{\#parameters}}} + \underbrace{L(\mathbf{d}|\boldsymbol{\theta}, m)}_{\substack{\text{neg-log-}\\\text{likelihood}}}$$

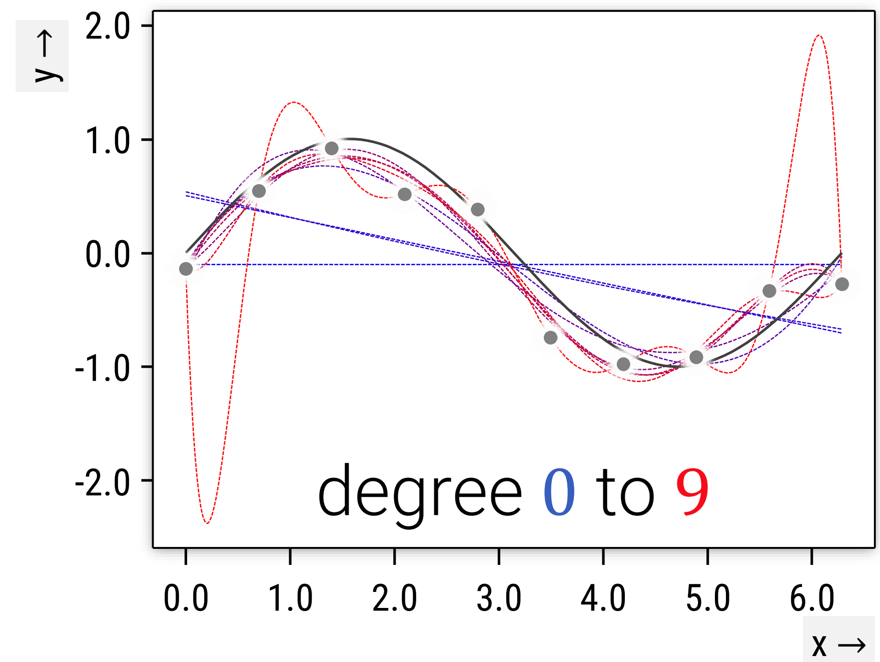
Example

Polynomial Regression

- Model m: Polynomial of degree $D = 0,1,2 \ldots, 9$
- Parameters $\boldsymbol{\theta}$ (for fixed m):
  - Coefficients in $\mathbb{R}^D$
  - Encoded in floating point: $O(D)$ bits
- Data $\mathbf{d}$: samples from function at n points
  - If model is good, no extra bits needed
  - If model is bad, many extra bits needed
  - Bad = uncertain or inaccurate
    - Both increase coding length
    - Uncertainty increases entropy
    - Inaccuracy asks for uncommon (long) codes

Model Selection Example

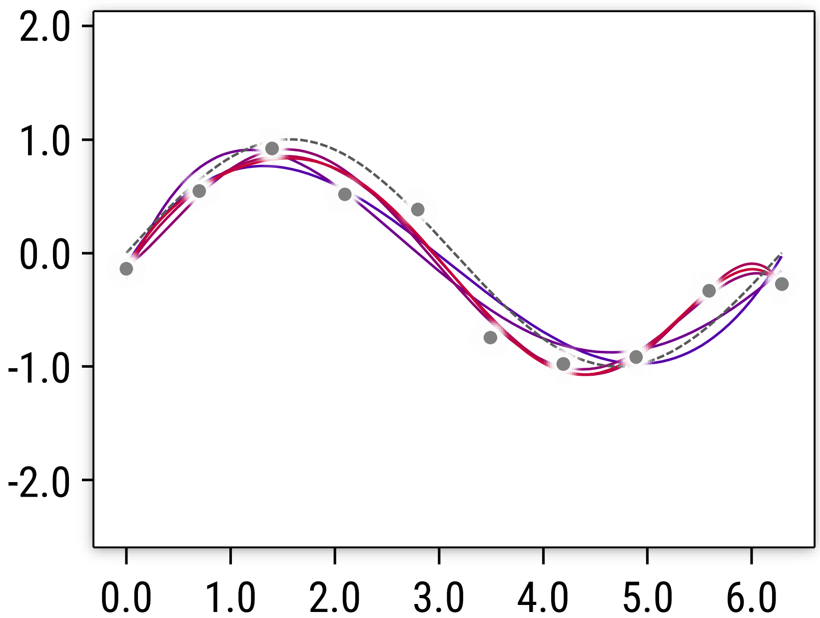




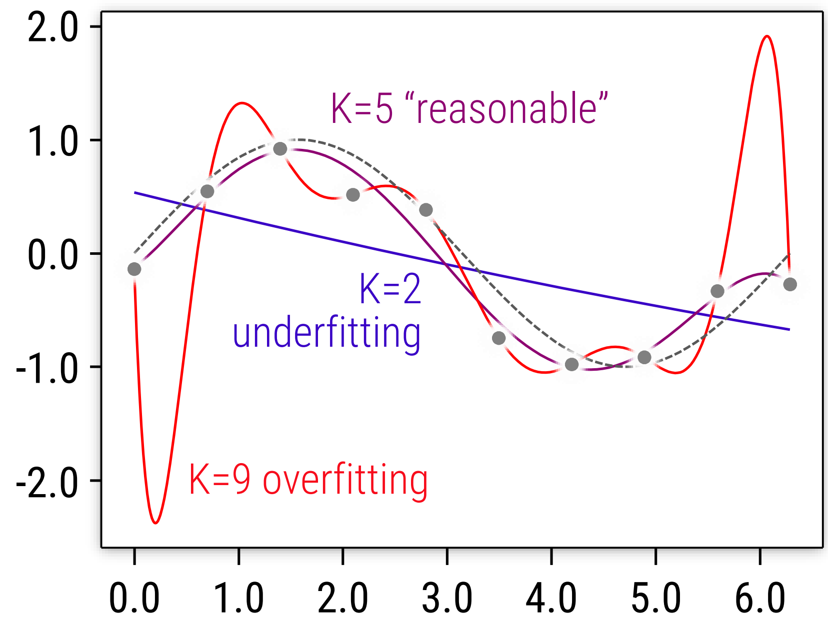
Polynomial approximation

- 10 samples from sine curve
- Approximation with polynomial of degree 0 to 9

Model Selection



Degree 3 to 7

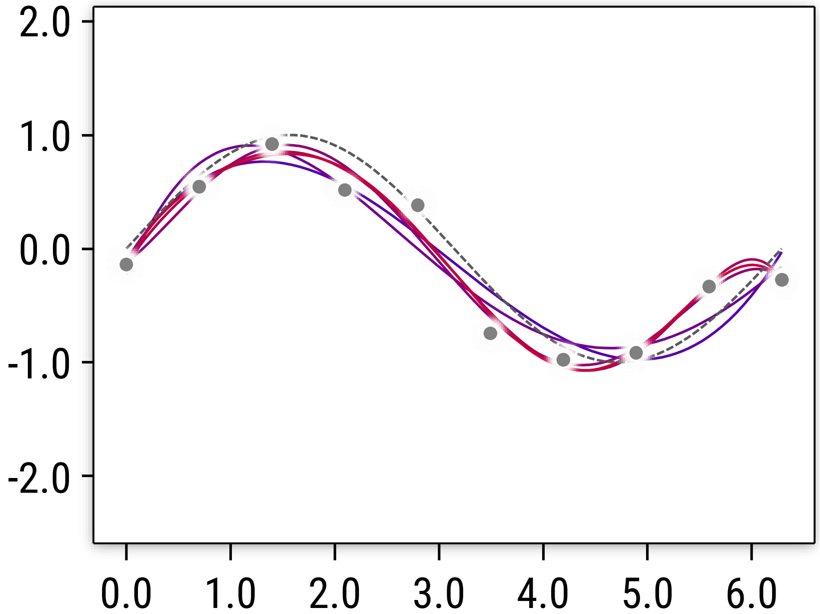


Degrees 2, 5, 9

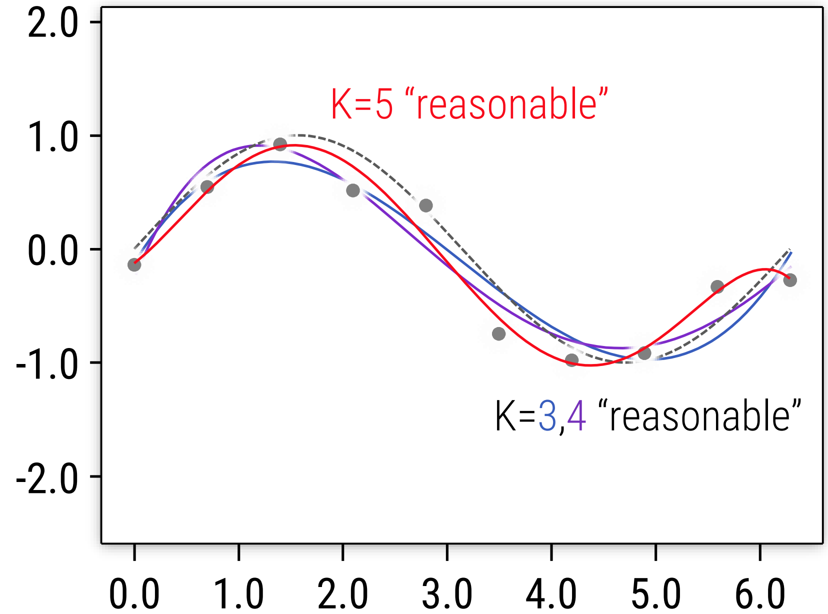
Empirically

- Degrees 3-7 "reasonable"
- Degree 5 closest fit, degree 3,4 less wiggly

Model Selection



Degree 3 to 7

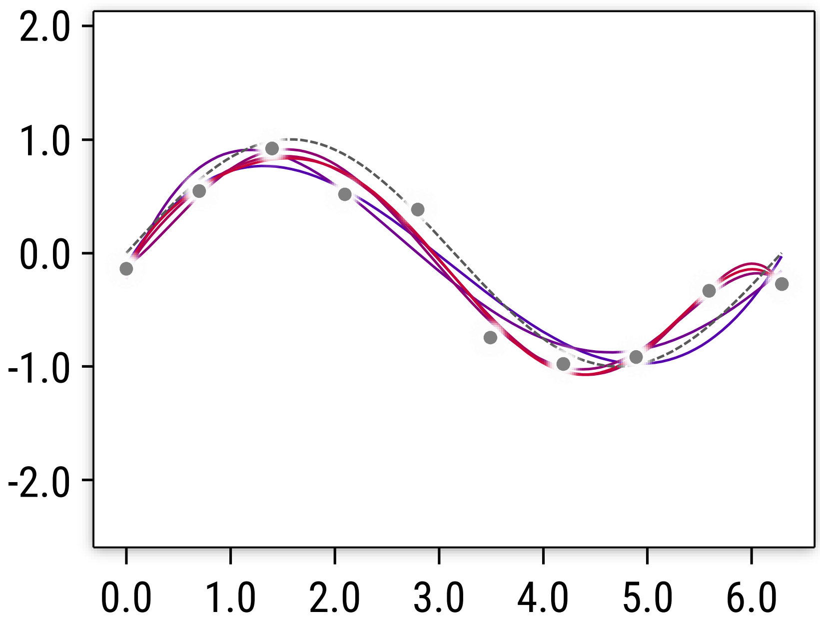


Degrees 3, 4, 5

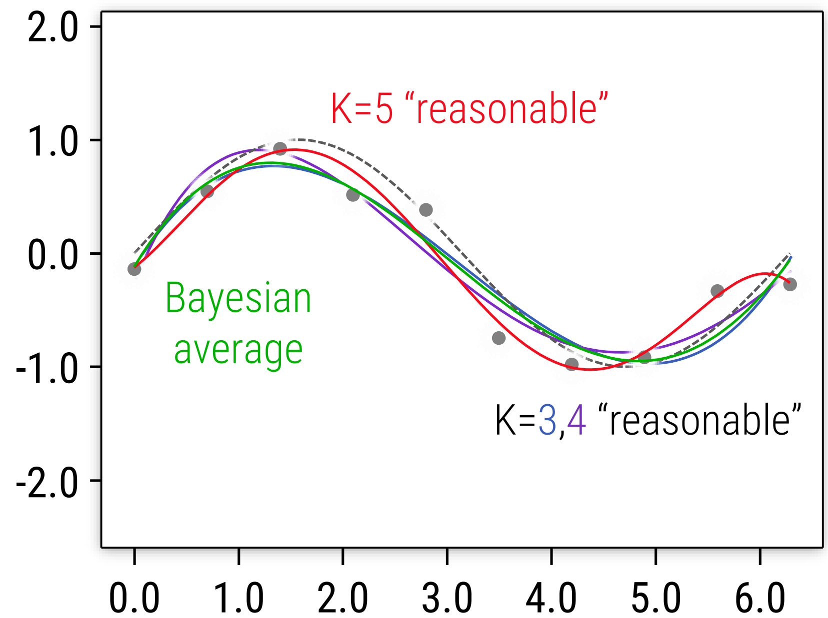
Empirically

- Degrees 3-7 “reasonable”
- Degree 5 closest fit, degree 3,4 less wiggly

Model Selection



Degree 3 to 7

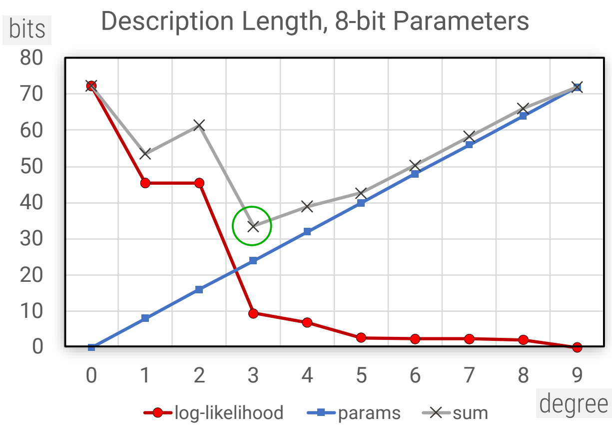


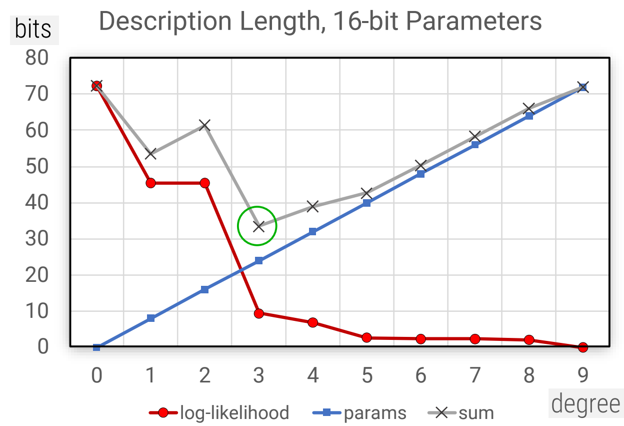
Degrees 3, 4, 5

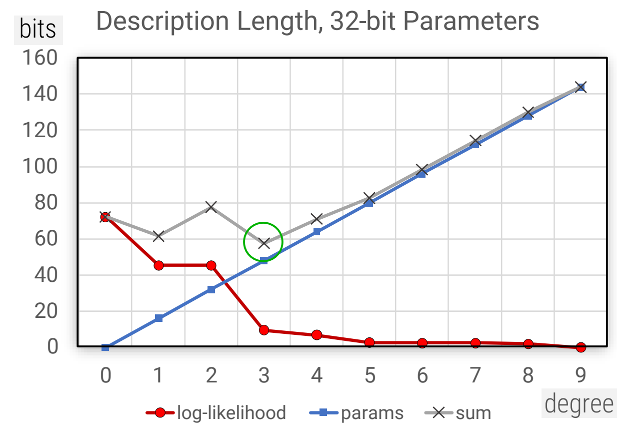
Empirically

- Degrees 3-7 “reasonable”
- Degree 5 closest fit, degree 3,4 less wiggly

Example

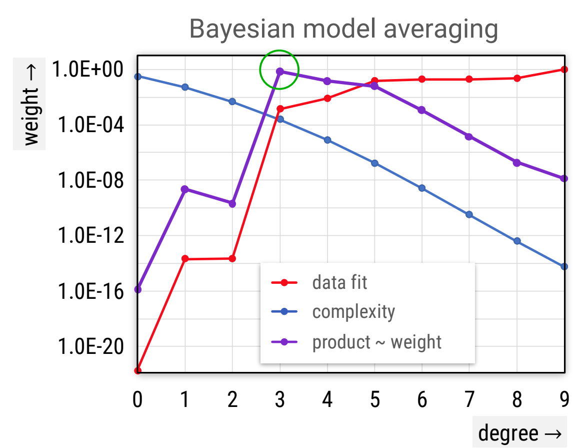






Polynomial Example

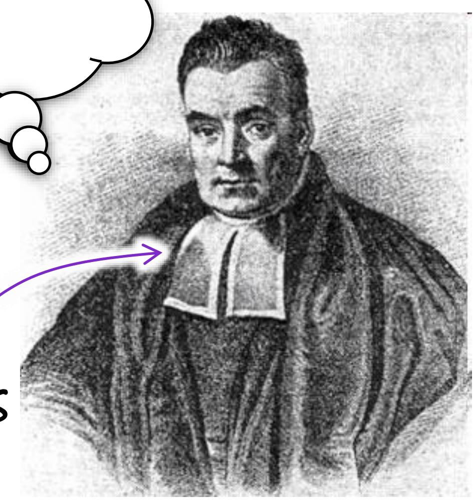
- Kind-of-works (degree 3 best)
- But discretization is still arbitrary for continuous parameters
- Need more "specific" entropy for $\boldsymbol{\theta}$



weighting

Bayesian Perspective on MDL

William of Ockham
(1287 – 1347)



Rev. Thomas Bayes
(c. 1701 – 1761)

[https://commons.wikimedia.org/wiki/File:William_of_Ockham_-_Logica_1341.jpg, https://en.wikipedia.org/wiki/Thomas_Bayes]

Bayesian Model Selection

Consider two variants

- “MAP-Style” MDL
  - Simple, but ad-hoc
- “Full-Bayesian” model selection
  - Relationship to / interpretation as MDL

MAP-Style MDL

Sending a model

- We fix a model m to assess
- Joint density:

$$P(\mathbf{d}, \boldsymbol{\theta} | m) = P(\mathbf{d} | \boldsymbol{\theta}, m) P(\boldsymbol{\theta} | m)$$

- Posterior for $\boldsymbol{\theta}$:

$$P(\boldsymbol{\theta} | \mathbf{d}, m) \sim P(\mathbf{d} | \boldsymbol{\theta}, m) P(\boldsymbol{\theta} | m)$$

- Determine $\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} P(\boldsymbol{\theta} | \mathbf{d}, m)$

Model Selection

- Compute message length for all m and pick shortest

MAP-Style MDL

$$\underbrace{enc(m)}_{\text{model prior}}\ \underbrace{enc(\boldsymbol{\theta})}_{\text{model complexity}}\ \underbrace{enc(d_1)\cdots enc(d_n)}_{\text{data fit}}$$

Sending a model

- Encode choice of m
  - Use $L(m) = -\log \mathrm{P}(m)$
  - Can encode a priori model preferences
- Encode parameters $\boldsymbol{\theta}$
  - Determine parameter prior $P(\boldsymbol{\theta}|m)$
  - $L(\widehat{\boldsymbol{\theta}}|m) = -\log P(\widehat{\boldsymbol{\theta}}|m)$ bits
- Encode data $\mathbf{d}$
  - $L(\mathbf{d}|\widehat{\boldsymbol{\theta}}, m) = -\log P(\mathbf{d}|\widehat{\boldsymbol{\theta}}, m)$
  - Neg-log-likelihood of best fitting model

Bayesian MDL

Bayesian Model Selection

- Inferring model

$$P(m|\mathbf{d}) = \frac{P(\mathbf{d}|m)P(m)}{P(\mathbf{d})}$$

$$\sim \underbrace{P(\mathbf{d}|m)}_{\substack{\text{marginal}\\ \text{likelihood}}} \underbrace{P(m)}_{\substack{\text{model}\\ \text{prior}}}$$

- We would select the most likely model m
  - Product of marginal likelihood and model prior
  - Reminder: Needs computation of marginal likelihood

$$P(\mathbf{d}|m) = \sum_{\boldsymbol{\theta}\in\Omega(\Theta)} P(\mathbf{d}|\boldsymbol{\theta}, m)P(\boldsymbol{\theta}|m) \quad \text{(which can be expensive)}$$

Integral for continuous Θ

Bayesian MDL

Bayesian Model

- We have so far…

$$P(m|\mathbf{d}) \sim \underbrace{P(\mathbf{d}|m)}_{\substack{\text{marginal}\\\text{likelihood}}} \underbrace{P(m)}_{\substack{\text{model}\\\text{prior}}}$$

- …and…

$$P(\mathbf{d}|m) = \sum_{\boldsymbol{\theta}\in\Omega(\boldsymbol{\Theta})} P(\mathbf{d}|\boldsymbol{\theta}, m)P(\boldsymbol{\theta}|m)$$

Encoding

- Send model: $L(m) = -\log P(m)$ (choose model m)
- Send data: $L(\mathbf{d}|m) = -\log P(\mathbf{d}|m)$ (send $\mathbf{d}$, model-based)

Bayesian MDL

$$\underbrace{enc(m)}_{\text{optionally variable}} \; \underbrace{enc(d_1) \cdots enc(d_n)}_{\text{data fit \& model complexity (implicit)}}$$

Encoding

- Model costs: $L(m) = -\log P(m)$
  - Prior for model selection
  - Optional/hand-tunable (in this context)
  - For non-uniform $P(m)$, this part is not constant
- Data costs: $L(\mathbf{d}|m) = -\log P(\mathbf{d}|m)$
  - Marginal likelihood gives direct encoding model for the data
  - Parameter costs are implicit (“1 part model”)

Bayesian MDL

$$\underbrace{enc(m)}_{\text{optionally variable}}\ \underbrace{enc(d_1)\cdots enc(d_n)}_{\text{data fit \& model complexity (implicit)}}$$

Encoding

- Tighter fit than “MAP-Style”
  - MAP-Style costs:

$$\min_{\boldsymbol{\theta}\in\Omega(\Theta)}\left(-\log P(\boldsymbol{\theta}|m) - \log P(\mathbf{d}|\boldsymbol{\theta}, m)\right)$$

$$= \min_{\boldsymbol{\theta}\in\Omega(\Theta)}\left(-\log P(\mathbf{d}|\boldsymbol{\theta}, m)P(\boldsymbol{\theta}|m)\right)$$

  - Bayes:

$$-\log\left(\sum_{\boldsymbol{\theta}\in\Omega(\Theta)} P(\mathbf{d}|\boldsymbol{\theta}, m)P(\boldsymbol{\theta}|m)\right)$$

- Bayesian expression is never larger
  - Tighter fit (better complexity estimate)

Note on MDL

There are more variants

- “Normalized Maximum Likelihood”
  - Theoretical advantages over Bayesian approach
- “Coarse” (Grunwald), ad-hoc MDL
  - Define approximate coding length along-the-way
- Common obstacle
  - Continuous variables carry infinite information
  - Address for example with accuracy constraints
    - See MacKay’s book Ch. 28
  - Bayesian method models noise in data explicitly

Bayesian Model Selection & Averaging

Model Selection

Bayesian approach

- Which model is better?
  - Model m_1 vs m_2
- Simple question
  - Compare $P(m_1|\mathbf{d})$ with $P(m_2|\mathbf{d})$
  - Select more likely

Fancy version: Bayesian model averaging

$$\bar{\boldsymbol{\theta}} = \int_{m \in \Omega(M)} \boldsymbol{\theta} \cdot P(m|\mathbf{d})\, d\boldsymbol{\theta}$$

- If models share parameters & params are vectors

Bayesian MDL

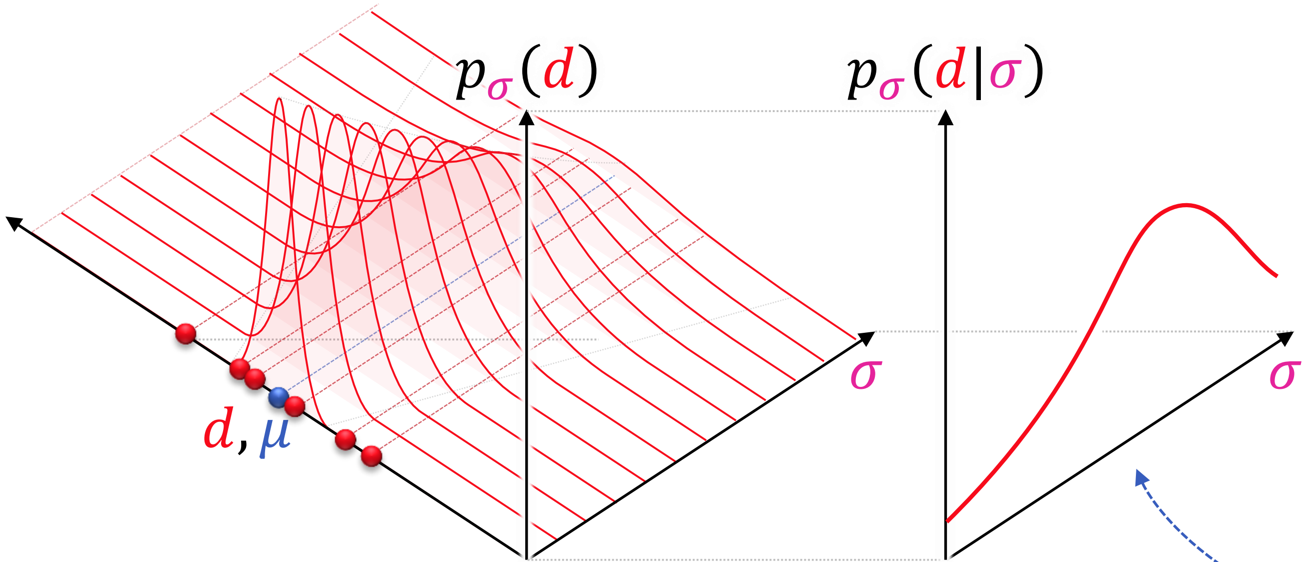
Bayesian Model Selection

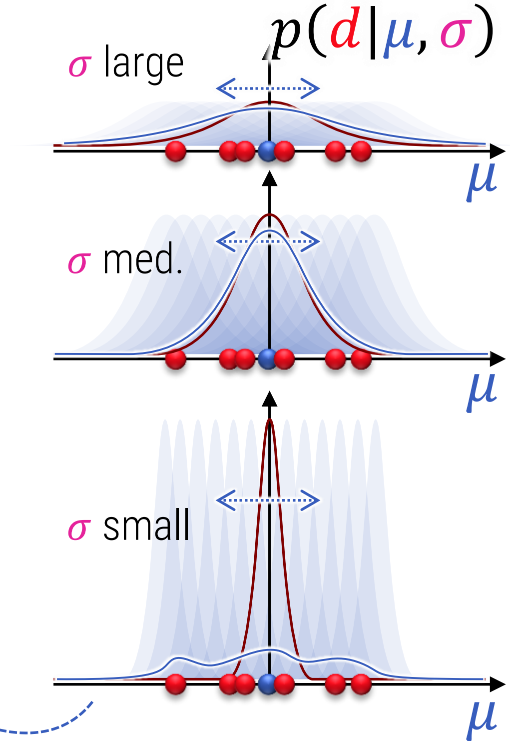
$$P(m|\mathbf{d}) = \frac{P(\mathbf{d}|m)P(m)}{P(\mathbf{d})}$$

$$\sim \underbrace{P(\mathbf{d}|m)}_{\substack{\text{marginal}\\\text{likelihood}}} \underbrace{P(m)}_{\substack{\text{model}\\\text{prior}}}$$

$$= \sum_{\boldsymbol{\theta} \in \Omega(\boldsymbol{\Theta})} \underbrace{\underbrace{P(\mathbf{d}|\boldsymbol{\theta}, m)}_{\substack{\text{data}\\\text{likelihood}}} \underbrace{P(\boldsymbol{\theta}|m)}_{\substack{\text{parameter}\\\text{prior}}}}_{\substack{\text{marginal}\\\text{likelihood}}} \underbrace{P(m)}_{\substack{\text{model}\\\text{prior}}}$$

Simple Example





Gaussian Model

$$p_\sigma(d) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(d-\mu)^2}{2\sigma^2}}$$

$$(\text{data } d, \underbrace{\text{mean } \mu}_{\text{parameter}}, \underbrace{\text{variance } \sigma}_{\text{“model”}})$$

Polynomial Fit

Reminder: Least-Squares Fit

- $f_{\mathbf{c}_K}(x_i) = \underbrace{\left(x_i^0, \dots, x_i^d, \dots, x_i^K\right)}_{\boldsymbol{\xi}_i^T} \cdot \mathbf{c}_K = \boldsymbol{\xi}_i^T \cdot \mathbf{c}_K$
- Design matrix $\mathbf{A} = \boldsymbol{\xi}_i \boldsymbol{\xi}_i^T$, optimum $\hat{\mathbf{c}}_K$

Marginal Likelihood

$$P(D|K) \sim \sigma_c^{-K} \cdot e^{-\frac{1}{2\sigma_D^2}\left(\sum_{i=1}^n y_i^2 - \hat{\mathbf{c}}_K^2\right)} \cdot \det\left(\mathbf{A} + \frac{\sigma_D^2}{\sigma_c^2}\mathbf{I}\right)^{-\frac{1}{2}}$$

Flat (improper) Prior

$$P(D|K) \sim \underbrace{e^{-\frac{1}{2\sigma_D^2}\left(\sum_{i=1}^n y_i^2 - \hat{\mathbf{c}}_K^2\right)}}_{\text{data fit}} \cdot \underbrace{\det(\mathbf{A})^{-\frac{1}{2}}}_{\text{complexity penalty}} \tag{143}$$

Connection to MDL

Gaussian Distributions

$$\mathcal{N}_{\mu,\boldsymbol{\Sigma}}(\mathbf{x}) := \left(\frac{1}{(2\pi)^{\frac{d}{2}}\det(\boldsymbol{\Sigma})^{\frac{1}{2}}}\right) e^{-\frac{1}{2}(\mathbf{x}-\mu)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\mu)}$$

d dimensions, cov. matrix $\boldsymbol{\Sigma}$

- Has (differential) entropy

$$\mathrm{H}\left(\mathcal{N}_{\mu,\boldsymbol{\Sigma}}\right) = \ln\left[(2\pi e)^{d}\det(\boldsymbol{\Sigma})^{\frac{1}{2}}\right]$$

"Complexity Penalty"

$$P(D|K) \sim \underbrace{e^{-\frac{1}{2\sigma_D^2}\left(\sum_{i=1}^{n} y_i^2 - \hat{\mathbf{c}}_K^2\right)}}_{\text{data fit}} \cdot \underbrace{\det(\mathbf{A})^{-\frac{1}{2}}}_{\text{complexity}}$$

$$= \underbrace{e^{-\frac{1}{2\sigma_D^2}\left(\sum_{i=1}^{n} y_i^2 - \hat{\mathbf{c}}_K^2\right)}}_{\text{data fit}} \cdot \underbrace{e^{-c\cdot H(\text{param})}}_{\text{complexity}} \quad (144)$$

entropy of data y_i under model (coding length)

entropy of parameters $\mathbf{c}_K$ (coding length)

Finite Resolution?

Version with prior

- Pentalty: $\det\left(\mathbf{A} + \frac{\sigma_D^2}{\sigma_C^2}\mathbf{I}\right)^{-\frac{1}{2}}$
- Ratio of
  - Noise in data (absolute precision)
  - Expected range of variability
- Regularizer – adding identity matrix limits resolution
  - Determinant is product of eigenvalues (main axis variances)
  - Singular if variance is zero in one direction
  - Sensitive to very small values
    - Identity creates “noise floor” at std.-dev. $\frac{\sigma_D}{\sigma_C}$
    - Below this, “nothing matters”

Perspectives

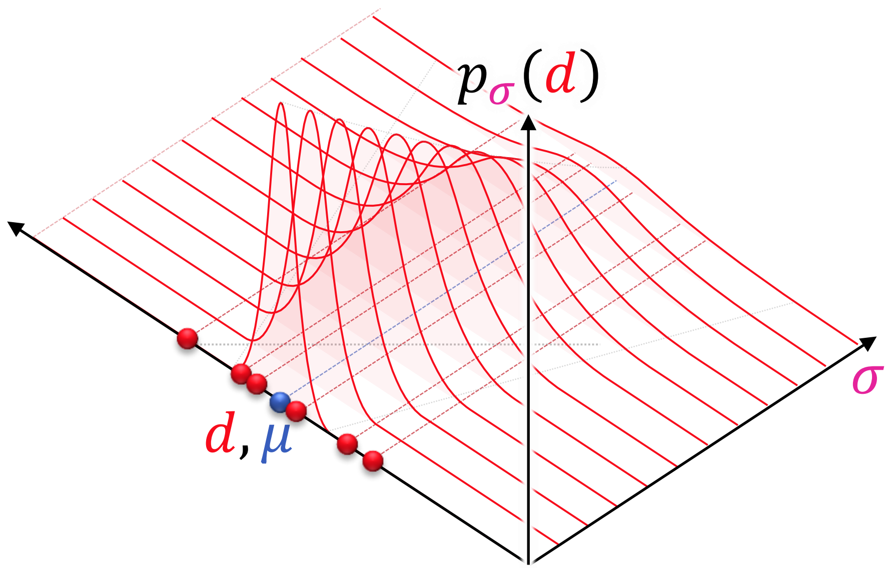
A Bit of Caution Needed…

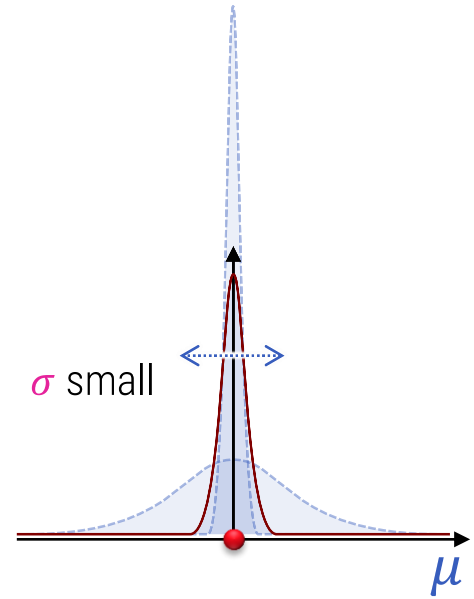
MDL & Model Selection

- Literature gives varying accounts
  - MDL just special case of Bayesian inference [MacKay 2003]
  - MDL more general [Grunwald 2004]
  - Arguments revolve around the role of priors (as often)
- Bayesian pitfalls
  - Bayesian model selection requires proper priors [MacKay 2003, Dawid et al. 1997]
  - It might work without (our example), but there are dragons

A.P. Dawid, M. Stone, J.V. Zidek:
Critique of E.T. Jaynes's "Paradox of Probability Theory", 2003
https://www.ucl.ac.uk/drupal/site_statistics/sites/statistics/files/rr172.pdf

Simple Example





singular for single sample

Gaussian Model

$$p_\sigma(d) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(d-\mu)^2}{2\sigma^2}}$$

$$(\text{data } d, \underbrace{\text{mean } \mu}_{\text{parameter}}, \underbrace{\text{variance } \sigma}_{\text{“model”}})$$

When to Use What?

Bayesian Averaging

- Marginal likelihood tractable (and good nerves)
- Estimating vectorial model parameters

Bayesian Model selection

- General model parameters

MDL (e.g., ad-hoc/MAP-Style)

- Marginal likelihood intractable (or too much for my nerves)

Frequentist generalization bounds

- Need guarantees on excess loss

When to Use What?

None of the above

- Simple model, tons of data (WCPGW, YOLO)
- Hand-tuned regularizer (e.g. MAP applications)
- Deep Networks (because, who knows)

When to Use What?

What should I do nonetheless?

- Use validation data
  - Separate from training data
- Monitor generalization performance
  - …during computational optimization
  - …during manual model-tuning
- Use test set, separated at the beginning
  - Use only once to measure generalization performance
  - Perform frequentist significance test
  - Report these numbers to your customer
    - Or scientific journal, if you are in that business
  - Manual overfitting to the test set is possible

Summary

Model Selection

Information Theory

- Intuitive arguments for

$$\#\text{data samples} \sim \#\text{model parameters}$$

- "Data sends us information through experiments"

Minimum Description Length

- Objective: compact encoding of the data
- "Best compression": model size + data size
- Roughly:

$$\min\,(\text{data neg log likelihood} + \text{parameter entropy})$$

Model Selection

Bayesian Model Selection

- Special case of probabilistic compression model
- Works well, but is technically “sensitive”
  - Marginal likelihood for comparison
    - Might be intractable
    - Might be nasty to compute even if tractable
  - Need to think seriously about
    - Priors
    - Error bars on data
- Bayesian model averaging strongly related
  - Just uses marginal likelihood as weight